NEW ESTIMATION ALGORITHMS WITH APPLICATIONS TO RELATIVE AND ABSOLUTE ORBIT DETERMINATION

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Shan Mohiuddin

May 2010

© 2010 Shan Mohiuddin

ALL RIGHTS RESERVED

NEW ESTIMATION ALGORITHMS WITH APPLICATIONS TO RELATIVE AND ABSOLUTE ORBIT DETERMINATION

Shan Mohiuddin, Ph.D.

Cornell University 2010

Three papers are presented that develop new estimation algorithms for the relative navigation of high-altitude spacecraft and for systems that have implicitly defined measurement times. The first two papers focus on the use of carrier-phase differential GPS techniques with integer ambiguities to estimate the relative states of spacecraft formations operating at altitudes above the GPS constellation. Those papers' contributions include an optimally integrated integer ambiguity factorization method for a square-root information estimator, a robust cycle-slip detection and recovery algorithm, and an extended Kalman filter with an adaptive ionosphere model that allows to make multimodal use of dual-frequency GPS data. The third paper develops a new process noise model and extended Kalman filter that sensibly processes implicitly constrained radio navigation measurements. The filter's models combine dense output Runge-Kutta numerical integration with a piecewise polynomial approximation to continuous-time white process noise.

BIOGRAPHICAL SKETCH

Shan Mohiuddin was born and raised in Wilmington, DE. He spent two years working as a boat builder in Newport, RI, before deciding to study engineering—first naval architecture, then aerospace—at Virginia Tech. After receiving a bachelor's degree, he interned at NASA/Goddard Space Flight Center where he contributed to the development of the Formation Flying Test Bed, a comprehensive GPS-based spacecraft simulation laboratory. During his graduate studies at Cornell, he was awarded a NASA Graduate Student Researchers Fellowship from 2004 to 2007. He has recently been hired by the Draper Laboratory in Cambridge, MA, where he will work as a GPS and navigation engineer.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Prof. Mark L. Psiaki, for his patient and skillful guidance. Thanks also to Russell Carpenter and Mike Moreau of NASA/Goddard Space Flight Center for supporting my research. Thanks to the many people who have helped and encouraged me over the years, especially Joe Hurley, Sonia Sylvanovich, Herdis Schopka, and above all, Charlotte Iorii. And finally, thanks to all my colleagues and friends in Ithaca who made my years here so memorable.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Spacecraft orbit determination, the field of engineering in which satellite trajectories are estimated and predicted, has both practical and scientific significance. Consider a practical application: GPS navigation. The GPS system, a radio navigation system that relies on accurate predictions of the broadcasting satellites' positions, allows millions of users worldwide to effortlessly and accurately determine their positions. Consider a scientific application: the TOPEX/Poseidon satellite mission that set out to map the ocean surface topology using radar altimetry. In order for the science data to be useful, however, the post-processed trajectory estimation error could be no more than 10 cm in the radial direction. The mission engineers met and exceeded the accuracy requirements and thus contributed directly the scientific success of the mission.

The successful orbit determination for these two examples relies on three elements: accurate modeling of the spacecraft's physical environment as it affects spacecraft motion, accurate modeling of the navigation measurements, and sensible data processing schemes. This dissertation attempts to contribute to these elements by developing new estimation algorithms for two types of problems. The first, the subject of Chapters 2 and 3, is the problem of high-altitude relative navigation. The second, the subject of Chapter 4, is the problem of processing implicitly constrained radio navigation measurements.

The paper presented in Chapter 2 extends the use of carrier-phase differential GPS (CDGPS) to altitudes above the GPS constellation. CDGPS relies on the precise but biased GPS carrier-phase measurements. The task of estimating the measurement biases, referred to as ambiguities, is eased under certain specific

circumstances by the fact they are guaranteed to be integer-values. The unique challenges found at high-altitudes, i.e., the poor signal strength and weak navigation geometry, brought into doubt that the ambiguities could be resolved as exact integers. The paper investigates this problem by developing a single-frequency pointwise estimator that optimally integrates an ambiguity resolution factorization technique. It also develops a robust carrier-phase cycle slip detection and recovery algorithm. The methods are demonstrated both with experimental data and simulated data.

The paper presented in Chapter 3 builds on the methods developed in Chapter 2 by deriving detailed dynamics and measurement models for a new extended Kalman filter. The filter's models allow it to forgo the typical a priori CDGPS data processing that throws away important information, thus giving the filter the flexibility to use the measurements in new ways. A new ionosphere model, for example, uses the geometry of the high-altitude scenarios to predict when the ionospheric effects will be significant and when they can be neglected. This knowledge allows the filter to make multimodal use of the dual-frequency CDGPS measurements. It adaptively chooses to use the dual-frequency information to directly correct for the ionospheric effects or to use it to aid in the resolution of the integer ambiguities. The filter is demonstrated in truth-model simulation.

The paper presented in Chapter 4 addresses the problem of processing measurements that have implicitly defined effectiveness times and thus require iterations to estimate these times during the Kalman filter's measurement update step. Dense output numerical integration is used to ease the computational burden of these iterations. In this context, the paper proposes an new process noise

model that is more appropriate than the typical zero-order hold model. It models the process noise as a piecewise polynomial. The statistics of the model are developed. A new extended Kalman filter combines dense output Runge-Kutta integration, implicit measurement iterations, and the new piecewise polynomial process noise model into a joint dynamic propagation/measurement update step that produces a posteriori state and process noise polynomial coefficient estimates. The filter is demonstrated by solving a simple tracking problem using data from a truth-model simulation.

The dissertation's conclusions are presented in Chapter 5.

CHAPTER 2

SATELLITE RELATIVE NAVIGATION USING CARRIER-PHASE DIFFERENTIAL GPS WITH INTEGER AMBIGUITIES

S. Mohiuddin and M.L. Psiaki, "High-Altitude Satellite Relative Navigation Using Carrier-Phase Differential Global Positioning System Techniques," *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 5, September-October 2007.

2.1 Abstract

A new carrier-phase differential GPS (CDGPS) relative navigation estimator has been developed that extends the use of CDGPS techniques to spacecraft formations that operate at geostationary altitudes and above. The estimator achieves rapid convergence to the carrier-phase ambiguities and incorporates a cycle slip detection and recovery algorithm. It solves a linearized problem using least-squares square-root information processing that does not require spacecraft dynamics models. In this context, integer ambiguities are resolved using an integer least-squares algorithm, e.g. LAMBDA. The cycle slip algorithm identifies the slip channel by statistical hypothesis testing and estimates the magnitude of the slip. GPS receiver-in-the-loop tests with simulated low-Earth orbit data show nearly instantaneous convergence to the correct integer ambiguities and relative position error magnitudes of less than 3 mm. Truth-model simulations are used to simulate geostationary orbits (GEO) and high-Earth orbit (HEO) scenarios. The GEO scenario produces nearly instantaneous convergence to the ambiguities and error magnitudes of less than 0.1 m. The HEO case at a radial distance of 17.8 Re converges in minutes with error magnitudes of less than 3 m.

Cycle slips, present in the hardware-in-the-loop simulations, are detected and corrected without significant accuracy degradation.

2.2 Introduction

Spacecraft formation flying is a key technology for future space flight. For many new mission concepts, e.g. NASA's Magnetosphere Multiscale Mission,[5] distributing the sensing over a formation offers both scientific and operational advantages over single-platform sensing. In order to achieve the benefits of distributed sensing, each spacecraft must have a high degree of autonomy, including the ability to autonomously navigate within the formation to a high degree of accuracy. Carrier-phase differential GPS (CDGPS) techniques are ideally suited for such accurate relative navigation. Low-Earth orbit (LEO) CDGPS systems have already produced cm- and sub-cm-level relative position accuracies using both simulated and flight data. The next step is to expand the application region to include geostationary orbits (GEO) and high-Earth orbits (HEO), both of which require the GPS receivers to operate above the GPS constellation. A CDGPS system operating in these high-altitude orbits must overcome poor visibility and poor geometric dilution of precision (GDOP), acquire and track very weak signals, and resolve carrier-phase measurement ambiguities with very slow receiver/GPS-satellite relative motion. Dual-frequency civilian technology may help solve these problems, but a practical system must be built upon a firm understanding of the capability of single-frequency CDGPS techniques. With this in mind, this paper examines and improves existing single-frequency CDGPS technology for space-based relative navigation.

An improved single-frequency system must make immediate use of newly acquired GPS signals, navigate continuously through periods of high GPS signal turnover, and detect and recover from carrier-phase cycle slips. These improvements are achievable with fast and reliable carrier-phase measurement ambiguity estimation techniques. Particular attention should be paid to incorporating double-differenced integer ambiguity resolution in an optimal manner. With the weaker observability of HEO and GEO systems, real-valued ambiguities would greatly increase the effective GDOP. This increase must be avoided since the systems are already challenged by poor geometry and noisy measurements. The proper use of double-differenced integer biases is, therefore, essential for successful high-altitude relative navigation systems.

Single-differenced CDGPS techniques, which result in real-valued ambiguities, have produced impressive accuracies in LEO simulations, often exhibiting 3-dimensional relative position errors of less than 3 mm over baselines distances of less than 10 km.[4] Double-differenced techniques have also proven accurate in LEO simulations, but some researchers have either estimated the double-differenced ambiguities as real numbers[13] or skirted the issue of how to begin the integer estimation process without a priori information.[28] These techniques are slow to converge and often retain uncertainty in the ambiguity estimates that can be eliminated by estimating the biases as exact integers. Other researchers have used optimal linear integer ambiguity estimation methods as part of dual-frequency CDGPS systems for LEO scenarios.[11,12] Such integer estimation methods are commonly called LAMBDA methods.[‡] All of these relative navigation

[‡]Least-square AMBiguity Decorrelation Adjustment (LAMBDA) solves a linear least-squares problem under the condition that elements of the solution vector are all integers. The algorithm calculates the ambiguities very efficiently by decorrelating the highly correlated ambiguities before performing a search to find the minimum cost vector estimate. The method is discussed further in the Solution Algorithm section (IV).

estimators were designed to operate in LEO where the GPS receivers will experience relatively high carrier-to-noise ratios, good navigation geometry, and fast line-of-sight vector dynamics, all of which increase the observability of the integer ambiguities. The existing algorithms that implement LAMBDA solve the integer problem in the context of the normal estimation equations, an approach that may encounter numerical problems under certain circumstances, and they make use of dual-frequency measurements. The estimator in Ref. 11 attempts to handle carrier-phase cycle slips by implementing a detection scheme, but it is unclear how it would recover from such events.

The present effort seeks to expand the applicability of CDGPS techniques. It has been motivated by the conjecture that a properly posed estimation problem and a carefully designed solution algorithm that optimally integrates LAMBDA-type methods would be able to resolve the carrier-phase integer ambiguities in the challenging operating environments in GEO and HEO using L1 measurements only. Once resolved, the relative position estimation error would only depend on the carrier-phase measurement error and the GDOP, resulting in theoretical accuracies of about 0.1 m in GEO and about 2 m in HEO.

This paper makes two main contributions. First, it presents a single-frequency relative navigation estimator that optimally integrates LAMBDA-type integer ambiguity resolution techniques into its solution algorithm in order to resolve the carrier-phase biases in high-altitude orbits. The estimator solves a linearized problem with least-squares square-root information processing, which is appropriate for use with the square-root information LAMBDA techniques in Ref. 22. The result is a simple formulation that does not require dynamics modeling, that is numerically stable, and that is easy to implement.

The estimator's ability to navigate in high-altitude orbits in the presence of poor navigation geometry, weak ambiguity observability, and noisy measurements is demonstrated using a truth-model simulation. The second contribution is the inclusion in the estimator of a robust carrier-phase cycle slip detection and recovery algorithm that uses statistical hypothesis testing to identify which channel has slipped, estimates the number of cycles that have slipped on that channel, and ensures that the magnitude of the slip settles to an integer value before the estimator reverts back to normal operation. It accomplishes the recovery without discarding any measurements. The technique is developed in the square-root information (SRI) format and is easily transferable to SRI filtering.

The remainder of the paper is divided into the following 6 major sections. Section II presents the measurement models, discusses the elements of the carrier-phase model that are important to relative navigation, and explains the single- and double-differenced carrier-phase measurements. Section III describes the linearized relative navigation problem. Section IV derives the solution algorithm, focusing on how it takes full advantage of the integer ambiguity assumption. Section V develops the slip detection scheme and the recovery algorithm. Section VI presents the results from simulations. Section VII provides a summary of the work and a discussion of its conclusions.

2.3 Carrier-Phase Model and Measurement Equations

The carrier-phase model presented in Ref. 23 forms the basis for the relative navigation algorithm. That model considers how the carrier signals are generated at the transmitters and how they are measured at the receivers, with a special

emphasis on how integer ambiguities arise in the context of CDGPS. Some of the relevant concepts are reviewed in this section, beginning with an overview of how a GPS receiver measures and reports carrier-phase. In subsequent paragraphs, single- and double-differenced measurements are discussed, focusing on the nature of the measurement ambiguities.

A GPS receiver produces its carrier-phase measurement by computing a beat phase, which is the phase difference between a replica of the nominal L1 carrier signal and the received carrier as determined by a phase-lock loop (PLL). The replica signal's phase equals an initial value $\gamma^j_{A_0}$ plus a linear function of receiver time whose slope equals the nominal L1 carrier frequency f_{L1}. Suppose that the tracked phase of the received signal, generated by the PLL, is ψ^j. Then the beat phase model is

$$\phi^j_{A_k} = \gamma^j_{A_0} + f_{L1} t_{A_k} - \psi^j_{A_k} \tag{2.1}$$

Note that superscripts indicate the tracked GPS satellite and subscripts indicate a particular receiver (A or B) and the measurement sample index (k). The last term in Eq. (2.1), the received phase at time t_{A_k}, can be modeled by considering the carrier signal broadcast by the GPS satellite and the signal propagation time. The received signal's phase $\psi^j_{A_k}$ equals an initial transmitted phase ψ^j_0 plus a linear function of the transmitter's clock time whose slope equals f_{L1}. This model is completed by considering the effects of the ionosphere, thermal noise, and most importantly, the difference between the receiver clock time and the transmitter clock time that is caused by the signal propagation time and the receiver and transmitter clock errors. Incorporating these terms into Eq. (2.1) and simplifying the expression yields the following carrier-phase measurement

equation:

$$\phi^j_{A_k} = \underbrace{\gamma^j_{A_0} - \psi^j_0}_{Ambiguity} + f_{L1}(\delta t_{A_k} - \delta t^j_k) + \frac{1}{\lambda_{L1}}\rho^j_{A_k} - I^j_{A_k} - \nu^j_{\phi_{A_k}} \tag{2.2}$$

In this equation, δt_{A_k} is the receiver clock error, δt^j_k is the satellite clock error, λ_{L1} is the nominal L1 carrier wavelength, $\rho^j_{A_k}$ is the receiver/GPS-satellite range, $I^j_{A_k}$ is an ionosphere effects term, and $\nu^j_{\phi_{A_k}}$ is the error caused by thermal noise and multipath. Notice how the measurement ambiguity is the difference between the initial phase of the receiver's nominal replica signal and the initial phase of the GPS satellite broadcast signal. In general, there is no reason to believe that the initial phases would be integers. The measurement ambiguity, therefore, is a real number.

Single-differencing of the measurements of two receivers tracking the same GPS satellite cancels out those terms that are common to that satellite, including the satellite broadcast signal's initial phase. It is assumed that the ionosphere errors are removed by this same process. For short baselines, this is a safe assumption because the line-of-sight vectors from the two receivers to a particular GPS satellite are nearly the same, resulting in ionosphere pierce points that are close to one another. For long baselines, this assumption breaks down. The subject of what to do for long baselines involves dual-frequency techniques that are beyond the scope of this paper. Please see Ref. 11 for more information on this topic. To simplify the notation, a single-differenced operator is defined as $\Delta(*)_{AB} = (*)_A - (*)_B$. Using this operator, the single-differenced carrier-phase becomes

$$\Delta(\phi)^j_{AB_{kl}} = \underbrace{\gamma^j_{A_0} - \gamma^j_{B_0}}_{\Delta Ambiguity} + f_{L1}\Delta(\delta t)_{AB_{kl}} + \frac{1}{\lambda_{L1}}\Delta(\rho)^j_{AB_{kl}} - \Delta(\nu)^j_{\phi_{AB_{kl}}} \tag{2.3}$$

The subscripts k and l indicates that the receiver measurement times t_{A_k} and t_{B_l} are not identical. If both receivers are applying their pseudorange-based navi-

gation solution clock correction terms, however, these times will agree to within the uncertainties of those corrections, or within about 100 ns, when four or more satellites are tracked at each receiver. For readability, the k and l subscripts will be suppressed for the remainder of the paper.

Notice that the ambiguities in Eq. (2.3) are still real numbers because they are differences of the real-valued initial phases of the nominal carrier replica signals in the two receivers. If the receivers were operating on the same local oscillator and if the receiver line-biases were calibrated out, then the single-differenced ambiguity would be an integer. These conditions are met in a typical attitude determination system, but in the case of relative navigation, the single-differenced ambiguities cannot be considered integers.

Taking single-differenced measurements for two different GPS satellites and differencing them produces a double-differenced measurement that cancels out those terms in Eq. (2.3) that are common to both receivers. The double-differenced operator is defined as $\nabla\Delta(*)^{ji}_{AB} = [(*)^j_A - (*)^j_B] - [(*)^i_A - (*)^i_B]$. Using this operator, the double-differenced phase is

$$\nabla\Delta(\phi)^{ji}_{AB} = \underbrace{\nabla\Delta(N)^{ji}_{AB}}_{\nabla\Delta Ambiguity} + \frac{1}{\lambda_{L1}}\nabla\Delta(\rho)^{ji}_{AB} + \nabla\Delta(\nu)^{ji}_{\phi_{AB}} \tag{2.4}$$

If a receiver is properly designed, then each channel uses the same nominal carrier-phase replica to compute the beat carrier phase, or it uses a replica that differs from the others by an integer number of cycles. This fact causes each difference $\gamma^j_{A_0} - \gamma^i_{A_0}$ and $\gamma^j_{B_0} - \gamma^i_{B_0}$ to be an integer number of cycles. The double-differenced ambiguity $\nabla\Delta(N)^{ji}_{AB} = (\gamma^j_{A_0} - \gamma^i_{A_0}) - (\gamma^j_{B_0} - \gamma^i_{B_0})$ is thus guaranteed to be an integer. Manipulating the data to achieve integers increases measurement uncertainty and computational overhead, but the savings in convergence time and calculation time in estimating the ambiguities as exact integers justify the

losses. Furthermore, once the integers have been estimated and validated, the relative navigation estimator can treat them as exact values, allowing the precise carrier-phase measurements to be treated as though they were very accurate pseudorange-like measurements of the relative position. The integer nature of the double-differenced ambiguities features prominently in the posing of the relative navigation problem and in the derivation of the solution algorithm.

It is appropriate to comment here on how a GPS receiver should be set up to work with a CDGPS system. Each receiver should report pseudorange and carrier-phase at its best estimate of the GPS second in order to ensure that the differencing operations do not introduce extra uncertainty as a result of large differences between the times t_{A_k} and t_{B_l}. Therefore, each receiver must produce an absolute navigation/clock solution and use the clock correction to guide its selection of sample times. Because it reports corrected times, the reported pseudorange must be modified to become $P_{corr} = P - c\delta t_0$, and the reported carrier-phase must be modified to become $\phi_{corr} = \phi - f_{L1}\delta t_0$.[9,23] Also, the availability of the pseudorange-based absolute solution for each receiver to the relative navigation system plays an important role in the linearization of the measurement equations.

2.4 Satellite Relative Navigation Problem

The target application for the present system is to achieve accurate relative navigation of 2 spacecraft in Earth orbit with arbitrary baseline distances. The algorithm must include fast ambiguity estimate convergence and reliable carrier-phase cycle slip detection and recovery. The system should operate with no

a priori knowledge of position and velocity. The following steps are taken to realize this system: (1) pose the estimation problem and linearize it in such a way that the integer nature of the double-differenced carrier-phase ambiguities is preserved, and (2) derive a solution algorithm that will ensure rapid convergence to the correct integer ambiguities and that can take advantage of computationally efficient decorrelation and integer search algorithms.

The estimation problem is posed as a linearized least-squares problem so that, when the integer condition of the double-differenced carrier-phase integer ambiguities is imposed, it can be solved with existing linear integer estimation techniques like the LAMBDA method. The nonlinear measurement equations are linearized about the pseudorange solutions at each receiver independently. One consequence of this linearization is that the position correction variable can be separated into two vectors: an absolute position correction and a relative position correction. The resulting relative position corrections can be smaller than the position corrections found in other linearization schemes, resulting in a more accurate linearized model and a more accurate estimate. The advantages of this separation will be discussed in more detail in the derivation of the linearization and the solution algorithm.

The posing of the relative navigation problem begins with the pseudorange measurement equations for the two receivers. The derivation is presented for receiver A, but the same steps apply to the measurement equations for receiver B. The subscripts indicating the measurement times are suppressed to simplify the notation.

$$P_A^j = \rho_A^j + c\delta t_A - c\delta t^j + \nu_{P_A}^j \tag{2.5}$$

P_A^j is the measured pseudorange, ρ_A^j is the true range, δt_A is the receiver clock

error, δt^j is the satellite clock error, and $\nu^j_{P_A}$ is the error caused by thermal noise and multipath. Linearizing the equations about the pseudorange solution and grouping the known terms on the left and the linear unknown terms on the right gives

$$P^j_A - c\delta t_{A_0} + c\delta t^j - \rho^j_{A_0} = (\hat{\rho}^j_A)^T \delta \mathbf{r}_A + c\partial t_A + \nu^j_{P_A} \tag{2.6}$$

The terms on the left side with subscript zero are the receiver clock correction and range computed by the pseudorange solution. On the right side, $\hat{\rho}^j_A$ is the unit direction vector that points from GPS satellite j to the pseudorange solution position for receiver A, $\delta \mathbf{r}_A$ is the absolute position correction vector to go from the pseudorange solution to the true position, and ∂t_A is the receiver clock correction increment to go from the pseudorange-based receiver clock correction to the true clock correction.

The CDGPS relative navigation algorithm uses modified unknowns in order to decompose the solution into a correction to the relative position, which it will determine precisely, and a correction to the absolute position. The receiver A absolute position correction vector can be represented as the sum of the receiver B absolute position correction vector $\delta \mathbf{r}_B$ and a relative position correction vector $\delta \mathbf{r}_{AB}$, thus separating the correction into two parts. A diagram of the change in unknowns is shown in Fig. 2.1, where $\mathbf{r}_{AB_0}$ is the differential position vector between the pseudorange solutions and $\mathbf{r}_{AB} = \mathbf{r}_{AB_0} + \delta \mathbf{r}_{AB}$ is the true differential position. The solution procedure will determine $\delta \mathbf{r}_{AB}$ and then add $\mathbf{r}_{AB_0}$ to it in order to compute $\mathbf{r}_{AB}$. Written in terms of these new unknowns, the linearized pseudorange equations for both receivers become

$$P^j_A - c\delta t_{A_0} + c\delta t^j - \rho^j_{A_0} = (\hat{\rho}^j_A)^T (\delta \mathbf{r}_B + \delta \mathbf{r}_{AB}) + c\partial t_A + \nu^j_{A_P} \tag{2.7}$$

$$P^j_B - c\delta t_{B_0} + c\delta t^j - \rho^j_{B_0} = (\hat{\rho}^j_B)^T \delta \mathbf{r}_B + c\partial t_B + \nu^j_{B_P} \tag{2.8}$$

The unknowns in the equations are $\delta\mathbf{r}_B$, $\delta\mathbf{r}_{AB}$, $c\partial t_A$, and $c\partial t_B$. If the carrier-phase data were not included, then the optimal solution for the quantities would be $\delta\mathbf{r}_B = \delta\mathbf{r}_{AB} = 0$ and $\partial t_A = \partial t_B = 0$. These would be the solutions because the pseudorange solutions already constitute the least-squares solutions to Eqs. (2.7) and (2.8) repeated for each tracked satellite.

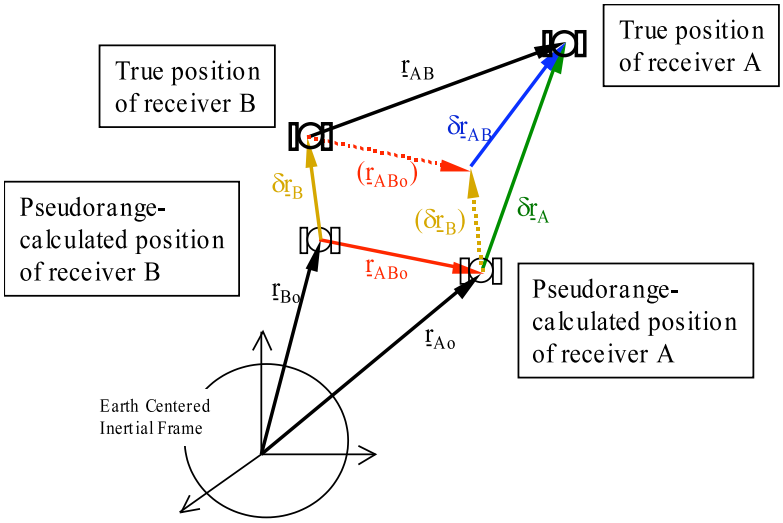

Figure 2.1: Modified relative navigation unknowns.

The absolute position correction vector $\delta\mathbf{r}_B$ can be large, on the order of the single-frequency pseudorange solution error of about 10 m or more. By explicitly separating the absolute and relative position correction terms, the relative position correction $\delta\mathbf{r}_{AB}$ is insulated from the large absolute position uncertainty associated with the pseudorange measurements. The more accurate relative position correction is the part of the solution that is most affected by the addition of the carrier-phase measurements. Separating the variables in this way is not strictly necessary, but is instructive in the context of deriving and analyzing the estimator. The equations include receiver clock correction increments to account for any remaining errors that are common to all channels in each receiver.

The same linearization scheme is applied to the double-differenced carrier-phase equation, which yields

$$\lambda_{L1}\nabla\Delta(\phi)^{ji}_{AB} - \nabla\Delta(\rho)^{ji}_{AB_0} = \lambda_{L1}\nabla\Delta(N)^{ji}_{AB} + [\nabla\Delta(\hat{\rho})^{ji}_{AB}]^T\delta\mathbf{r}_B + (\hat{\rho}^j_A - \hat{\rho}^i_A)^T\delta\mathbf{r}_{AB} + \nabla\Delta(v)^{ji}_{AB_\phi} \tag{2.9}$$

Again, the absolute and relative correction terms are separated. This equation has been manipulated to group the known terms on the left and the linear unknown terms on the right. As in Eqs. (2.7) and (2.8), the unknown terms associated with the absolute and relative position corrections are $\delta\mathbf{r}_B$ and $\delta\mathbf{r}_{AB}$. The third unknown term is the double-differenced integer ambiguity $\nabla\Delta(N)^{ji}_{AB}$.

2.5 Solution Algorithm

The linearized measurement equations are set up to form a least-squares problem that allows pointwise solutions for the absolute position, relative position, and clock corrections, and that allows sequential solutions for the double-differenced integer ambiguities. The problem is solved with a square-root information estimator, which keeps track of the square-root of the inverse of the ambiguity estimate covariance matrix. It carries this matrix forward, updating it at each time step, while solving from scratch for the updated relative and absolute positions at each time step. This type of estimator is typically called a real-time kinematic estimator. The square-root information form is numerically stable and suitable to work with the LAMBDA method. Both pseudorange and carrier-phase measurements are used in the solution. The inclusion of the pseudoranges limits the double-differenced carrier-phase ambiguity search volume to within the error of the pseudorange solution. This relatively small search

volume is further reduced by the imposition of the integer condition, shrinking the set of candidates from all the possible combinations of real-valued ambiguities in the search volume to all the possible combinations of integer-valued ambiguities in the search volume. Only a few viable integer sets exist within the pseudorange-limited search volume. This reduced search volume will promote fast convergence to the correct bias estimates and, when coupled with the LAMBDA method, will promote fast calculation of the estimates.

The solution algorithm begins by combining linearized pseudorange and carrier-phase measurements into two matrix-vector equations. The pseudorange equation for the system is developed by repeating Eqs. (2.7) and (2.8) for all tracked GPS satellites from 1 to J

$$\underbrace{\begin{bmatrix} z^1_{P_A} \\ z^2_{P_A} \\ \vdots \\ z^J_{P_A} \\ z^1_{P_B} \\ z^2_{P_B} \\ \vdots \\ z^J_{P_B} \end{bmatrix}}_{\mathbf{z}_{P_k}} = \underbrace{\begin{bmatrix} (\hat{\rho}^1_A)^T & (\hat{\rho}^1_A)^T & 1 & 0 & 0 & \dots & 0 \\ (\hat{\rho}^2_A)^T & (\hat{\rho}^2_A)^T & 1 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ (\hat{\rho}^J_A)^T & (\hat{\rho}^J_A)^T & 1 & 0 & 0 & \dots & 0 \\ 0 & (\hat{\rho}^1_B)^T & 0 & 1 & 0 & \dots & 0 \\ 0 & (\hat{\rho}^2_B)^T & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & (\hat{\rho}^J_B)^T & 0 & 1 & 0 & \dots & 0 \end{bmatrix}}_{[H_{x_{P_k}} \;\; 0]} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{N} \end{bmatrix} + \nu_{P_k} \tag{2.10}$$

where

$$\mathbf{x}_k = \begin{bmatrix} \delta\mathbf{r}_{AB_k} \\ \delta\mathbf{r}_{B_k} \\ c\partial t_{A_k} \\ c\partial t_{B_k} \end{bmatrix} \qquad \mathbf{N} = \begin{bmatrix} \nabla\Delta(N)^{21}_{AB} \\ \nabla\Delta(N)^{31}_{AB} \\ \vdots \\ \nabla\Delta(N)^{J1}_{AB} \end{bmatrix} \tag{2.11}$$

$$z^j_{P_A} = P^j_A - \rho^j_{A_0} + c(\delta t^j - \delta t_{A_0}) \tag{2.12}$$

$$z^{j}_{P_B} = P^{j}_{B} - \rho^{j}_{B_0} + c(\delta t^{j} - \delta t_{B_0}) \tag{2.13}$$

and ν_{P_k} is the pseudorange measurement noise vector. Note that $H_{x_{P_k}}$ consists of the first 8 columns of the large matrix on the right hand side of Eq. (2.10).

The carrier-phase equation for the system is developed by repeating Eq. (2.9) for all GPS satellite pairs $ji = 21, 31, ..., J1$

$$\underbrace{\begin{bmatrix} z^{21}_{\phi_A} \\ z^{31}_{\phi_A} \\ \vdots \\ z^{J1}_{\phi_A} \end{bmatrix}}_{\mathbf{z}_{\phi_k}} = \underbrace{\begin{bmatrix} [\hat{\rho}^{2}_{A} - \hat{\rho}^{1}_{A}]^T & [\nabla\Delta(\rho^{21}_{AB})]^T & 0 & 0 & \lambda_{L1} & 0 & \dots & 0 \\ [\hat{\rho}^{3}_{A} - \hat{\rho}^{1}_{A}]^T & [\nabla\Delta(\rho^{31}_{AB})]^T & 0 & 0 & 0 & \lambda_{L1} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ [\hat{\rho}^{J}_{A} - \hat{\rho}^{1}_{A}]^T & [\nabla\Delta(\rho^{J1}_{AB})]^T & 0 & 0 & 0 & 0 & 0 & \lambda_{L1} \end{bmatrix}}_{[H_{x_{\phi_k}} \quad H_{N_{\phi_k}}]} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{N} \end{bmatrix} + \nu_{\phi_k} \tag{2.14}$$

where

$$z^{ji}_{\phi} = \lambda_{L1}\nabla\Delta(\phi)^{ji}_{AB} - \nabla\Delta\rho^{ji}_{AB_0} \tag{2.15}$$

and ν_{ϕ_k} is the carrier-phase measurement noise vector. The matrix $H_{x_{\phi_k}}$ consists of the first 8 columns of the large matrix on the right hand side of Eq. (2.14), and $H_{N_{\phi_k}}$ consists of the remaining $J - 1$ columns.

The systems of linear equations in Eqs. (2.10) and (2.14) can be combined to yield

$$\begin{bmatrix} \mathbf{z}_{P_k} \\ \mathbf{z}_{\phi_k} \end{bmatrix} = \begin{bmatrix} H_{x_{P_k}} & 0 \\ H_{x_{\phi_k}} & H_{N_{\phi_k}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{N} \end{bmatrix} + \begin{bmatrix} \nu_{P_k} \\ \nu_{\phi_k} \end{bmatrix} \tag{2.16}$$

The ambiguity vector $\mathbf{N}$ is assumed to be constant over all measurement epochs. This is only true if no carrier-phase cycle slips occur. These occurrences will instantly degrade the accuracy of the relative position estimate and must be detected before the estimator reports the degraded position corrections. Such a detection system is developed in the next section.

Since the ambiguities are constant in the absence of cycle slips, it is useful to carry information from one sample to the next. In fact, the estimation problem at sample k uses all available information about the ambiguities from samples 0 through $k-1$. This information takes the form of the square-root information equation $\hat{R}_{NN_{k-1}}\mathbf{N} = \hat{\mathbf{z}}_{N_{k-1}} - \nu_{N_{k-1}}$, where $\hat{\mathbf{z}}_{N_{k-1}}$ is the a priori information vector, $\nu_{N_{k-1}}$ is the associated zero-mean, unit-variance Gaussian noise vector, and $\hat{R}_{NN_{k-1}}$ is the a priori ambiguity square-root information matrix (SRIM). All the elements in $\hat{\mathbf{z}}_{N_{k-1}}$ and $\hat{R}_{NN_{k-1}}$ are initially set to zero for $k = 0$ to indicate that no a priori information is available. Augmenting the system in Eqn. (2.16) with the a priori information gives

$$\begin{bmatrix} \mathbf{z}_{P_k} \\ \mathbf{z}_{\phi_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} = \begin{bmatrix} H_{x_{P_k}} & 0 \\ H_{x_{\phi_k}} & H_{N_{\phi_k}} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{N} \end{bmatrix} + \begin{bmatrix} \nu_{P_k} \\ \nu_{\phi_k} \\ \nu_{N_{k-1}} \end{bmatrix} \tag{2.17}$$

Square-root information processing requires that the measurement equations be pre-conditioned in order to make their zero-mean, Gaussian errors have unit covariances. This conditioning involves left multiplication by a square-root of the inverse covariance of the measurement errors in the original equations. The measurement error standard deviations are assumed to be $\sigma_P = 7$ m for the pseudorange measurements and $\sigma_\phi = 0.005$ m for the carrier-phase measurements. The pseudorange and carrier-phase measurements are independent on each channel, but when double-differenced, the carrier-phase errors are not independent. The following are the measurement error covariance matrices and their square-root inverse factorizations for the pseudorange and double-

differenced carrier-phase equations

$$\Sigma_P = \sigma_P^2 \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & & \vdots \\ \vdots & & \ddots & \\ 0 & \dots & & 1 \end{bmatrix} = R_{\nu\nu_P}^{-1}(R_{\nu\nu_P}^{-1})^T \tag{2.18}$$

$$\Sigma_\phi = \sigma_\phi^2 \begin{bmatrix} 4 & 2 & \dots & 2 \\ 2 & 4 & & \vdots \\ \vdots & & \ddots & \\ 2 & \dots & & 4 \end{bmatrix} = R_{\nu\nu_\phi}^{-1}(R_{\nu\nu_\phi}^{-1})^T \tag{2.19}$$

$R_{\nu\nu_P}$ equals σ_P^{-1} times an identity matrix and $R_{\nu\nu_\phi}$ can be computed from Σ_ϕ by inverting and transposing its Cholesky factorization. The problem is transformed to yield the desired square-root information form through left multiplication of the appropriate blocks by $R_{\nu\nu_P}$ and $R_{\nu\nu_\phi}$

$$\begin{bmatrix} \tilde{\mathbf{z}}_{P_k} \\ \tilde{\mathbf{z}}_{\phi_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} = \begin{bmatrix} R_{\nu\nu_P}\mathbf{z}_{P_k} \\ R_{\nu\nu_\phi}\mathbf{z}_{\phi_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} \tag{2.20}$$

$$\begin{bmatrix} \tilde{H}_{x_{P_k}} & 0 \\ \tilde{H}_{x_{\phi_k}} & \tilde{H}_{N_{\phi_k}} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} = \begin{bmatrix} R_{\nu\nu_P}H_{x_{P_k}} & 0 \\ R_{\nu\nu_\phi}H_{x_{\phi_k}} & R_{\nu\nu_\phi}H_{N_{\phi_k}} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} \tag{2.21}$$

$$\tilde{\nu}_k = \begin{bmatrix} R_{\nu\nu_P}\nu_{P_k} \\ R_{\nu\nu_\phi}\nu_{\phi_k} \\ \nu_{N_{k-1}} \end{bmatrix} \tag{2.22}$$

The solution algorithm then factorizes the coefficient matrix from Eq. (2.21) into the product of an orthonormal matrix and an upper triangular matrix (QR

factorization).[2,8] The result is

$$Q_k \begin{bmatrix} \hat{R}_{xx_k} & \hat{R}_{xN_k} \\ 0 & \hat{R}_{NN_k} \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} \tilde{H}_{x_{P_k}} & 0 \\ \tilde{H}_{x_{\phi_k}} & \tilde{H}_{N_{\phi_k}} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} \tag{2.23}$$

where Q_k is an orthonormal matrix, $\hat{R}_{xx_k}$ is an 8×8 upper triangular matrix, $\hat{R}_{NN_k}$ is a $(J-1) \times (J-1)$ upper triangular matrix, and $\hat{R}_{xN_k}$ is an $8 \times (J-1)$ dense matrix. These matrices can be determined from the block matrix on the right-hand-side of Eq. (2.23) using standard QR techniques. The measurement vector is transformed by left multiplying by the transpose of the Q_k matrix, giving

$$\begin{bmatrix} \hat{\mathbf{z}}_{x_k} \\ \hat{\mathbf{z}}_{N_k} \\ \mathbf{z}_{res_k} \end{bmatrix} = Q_k^T \begin{bmatrix} \tilde{\mathbf{z}}_{P_k} \\ \tilde{\mathbf{z}}_{\phi_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} \tag{2.24}$$

resulting in the transformed block upper-triangular system

$$\begin{bmatrix} \hat{\mathbf{z}}_{x_k} \\ \hat{\mathbf{z}}_{N_k} \\ \mathbf{z}_{res_k} \end{bmatrix} = \begin{bmatrix} \hat{R}_{xx_k} & \hat{R}_{xN_k} \\ 0 & \hat{R}_{NN_k} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{N} \end{bmatrix} + \tilde{\nu}_k \tag{2.25}$$

where $\tilde{\nu}_k$ is a zero-mean, unit-variance noise vector.

At this point, estimates could be derived by setting $\tilde{\nu}_k$ to zero and solving for $\mathbf{x}_k$ and $\mathbf{N}$ by back substitution. This approach, however, ignores the fact that $\mathbf{N}$ is a vector of integers. In order to take advantage of this knowledge, an integer ambiguity resolution step is carried out before back substitution is used to determine $\mathbf{x}_k$. This resolution is accomplished by passing the decoupled ambiguity SRIM $\hat{R}_{NN_k}$ and the associated information vector $\hat{\mathbf{z}}_{N_k}$ to the LAMBDA solution algorithm.

The LAMBDA method is designed to produce viable integer ambiguity candidate vectors and to minimize a cost function in order to find the best vector. It solves the following integer linear least-squares (ILLS) problem: find $\mathbf{N}$ to minimize

$$J(\mathbf{N}) = \frac{1}{2}(\hat{R}_{NN_k}\mathbf{N} - \hat{\mathbf{z}}_{N_k})^T(\hat{R}_{NN_k}\mathbf{N} - \hat{\mathbf{z}}_{N_k}) \tag{2.26}$$

under the condition that

$$\mathbf{N} \in Z^{J-1} \tag{2.27}$$

where Z^{J-1} indicates the space of $(J-1)$-dimensional integer vectors. The algorithm operates on the ambiguity SRIM $\hat{R}_{NN_k}$ using a unimodal linear transformation of the integers in order to approximately decorrelate the highly correlated ambiguities.[22] The transformed SRIM is passed to an integer search algorithm. That algorithm starts by generating a reasonable first guess of $\mathbf{N}_{opt}$, and it uses that guess to determine a cost that defines a bounded search volume. The algorithm finds all integer vectors inside the search volume, evaluates their costs, and designates the lowest cost vector as the correct ambiguity estimate.

The advantages of using this decorrelation and integer estimation algorithm are illustrated in Fig. 2.2. The upper plot shows the computed standard deviations for real-valued ambiguity estimates over a 600 s sample period. The standard deviations take more than 420 s to all drop below a unit value and would take more than 600 s to become small enough that the estimates could be safely rounded to the nearest integers. When the LAMBDA method is applied to this scenario, only a handful of valid integer ambiguity vectors are produced in the first time steps. After just 3 s, the number of valid candidates drops to 1, indicating to a high degree of probability that the correct integer vector has been identified. This behavior is indicated in the lower plot of Fig. 2.2. The LAMBDA method can resolve the ambiguities to their correct integer values

200 times faster than does an algorithm that estimates the ambiguities as real values and then rounds them to integers.

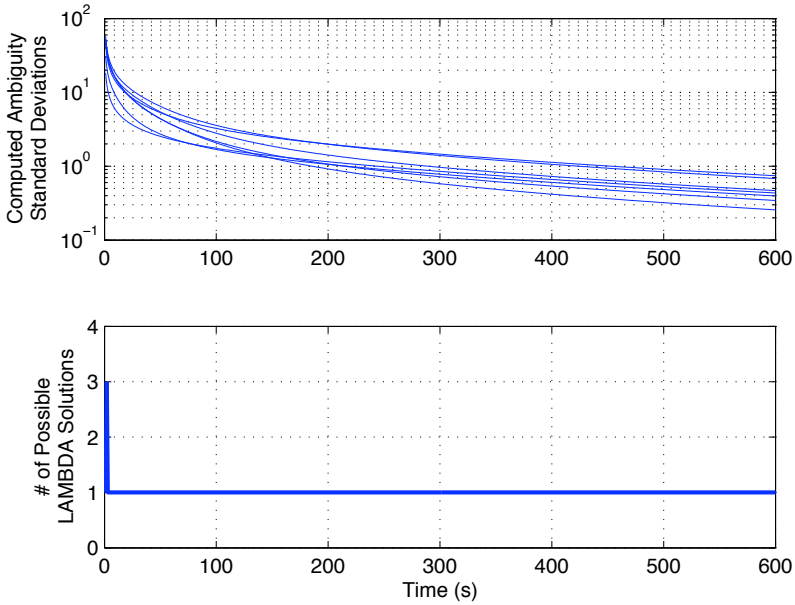

Figure 2.2: Comparison of real-valued ambiguity convergence and the number of valid integer ambiguity vectors produced by the LAMBDA method.

Once the integer ambiguity vector $\mathbf{N}_{opt}$ has been estimated, it is passed back to the relative navigation estimator, which uses it to solve for the position corrections:

$$\mathbf{x}_{opt_k} = \hat{R}_{xx_k}^{-1}(\hat{\mathbf{z}}_{x_k} - \hat{R}_{xN_k}\mathbf{N}_{opt}) \tag{2.28}$$

To complete the algorithm's recursion, the square-root information vector associated with the ambiguities $\hat{\mathbf{z}}_{N_k}$ and the ambiguity SRIM $\hat{R}_{NN_k}$ are passed forward to sample index $k+1$ as a priori information.

The executive solution algorithm is summarized in the following steps:

1. Set counter $k = 0$ and set diffuse prior for the ambiguity estimate SRIM $\hat{R}_{NN_k} = 0$ and the vector $\hat{\mathbf{z}}_k = 0$.

2. Update counter $k = k+1$, calculate the pseudorange solution, derive quantities for the transformed linearized measurement equations that appear in Eqs. (2.20) and (2.21), and QR factorize the problem as in Eqs. (2.23) and (2.24) in order to produce blocks for position corrections and ambiguity estimates.

3. Check for cycle slips. (The method for detecting cycle slips is described in the next section.) If yes, then apply slip recovery algorithm (also described in the next section). If no, continue.

4. Solve the ILLS problem associated with Eq. (4.59) to determine $\mathbf{N}_{opt}$.

5. Solve for position/clock corrections using Eq. (2.28) and add relative position correction to the relative position estimate calculated by differencing the pseudorange solution absolute positions.

$$\hat{\mathbf{r}}_{AB_k} = \mathbf{r}_{A_{P_k}} - \mathbf{r}_{B_{P_k}} + \delta r_{AB_k} \tag{2.29}$$

6. Retain ambiguity SRIM $\hat{R}_{NN_k}$ and the vector $\hat{\mathbf{z}}_{N_k}$. Go to Step 2.

2.6 Cycle Slip Detection and Recovery

The estimation process in the executive algorithm breaks down when the PLL of one of the receivers' channels produces a carrier-phase jump, commonly called a cycle slip. In this paper, the term carrier-phase measurement jump and carrier-phase cycle slip have similar meanings. The distinction, where applicable, is that a jump may not be an exact integer number of cycles. The term cycle slip tends to imply an integer number. A well designed receiver will experience few jumps, and the jumps will have non-integer values only during brief transients. After the transients, the slip stabilizes at a constant, exact integer.

Since the ambiguity estimates should be constant, a cycle slip is caught by looking for discrepancies between the a priori information and the current measurements. Such discrepancies appear as large jumps in the cost associated with the measurement residuals. Once the cost exceeds a threshold, the algorithm enters the Cycle Slip Recovery (CSR) mode. In this mode, it performs hypothesis testing to identify the slip channel, and it estimates the number of cycles that have slipped on that channel. Measurement errors of this type usually occur over several time steps, and although the errors will ultimately settle to integer numbers of cycles, they are treated as real numbers during the transients. Once the cycle slip estimate has converged to an integer, the algorithm reverts to the normal mode of operation.

The residual cost that is an indicator of measurement errors is defined as

$$J_{res} = \frac{1}{2}(\hat{R}_{NN_k}\mathbf{N}_{opt} - \hat{\mathbf{z}}_{N_k})^T(\hat{R}_{NN_k}\mathbf{N}_{opt} - \hat{\mathbf{z}}_{N_k}) + \frac{1}{2}\mathbf{z}_{res_k}^T\mathbf{z}_{res_k} \tag{2.30}$$

where the first term on the right-hand-side is the optimal LAMBDA cost from Eq. (4.59) and the second term is the cost associated with the residual least-squares error vector $\mathbf{z}_{res_k}$ from Eq. (2.24). The quantity $2J_{res}$ should be distributed like a Chi-squared distribution of degree $4J - 10$, provided there are no cycle slips. This distribution can be used to calculate a cost threshold that will indicate phase jumps. The calculation is carried out by solving an inverse cumulative distribution function with a user-specified false alarm probability. For a particular scenario, a reasonable false alarm probability results in a cost threshold of 30. Figure 2.3 shows typical cost values during good measurements and the increment in cost that occurs at a cycle slip.

Once a cycle slip is detected, the algorithm saves the last good ambiguity estimate as $\mathbf{N}_{old}$ for later comparison and then attempts to isolate the channel

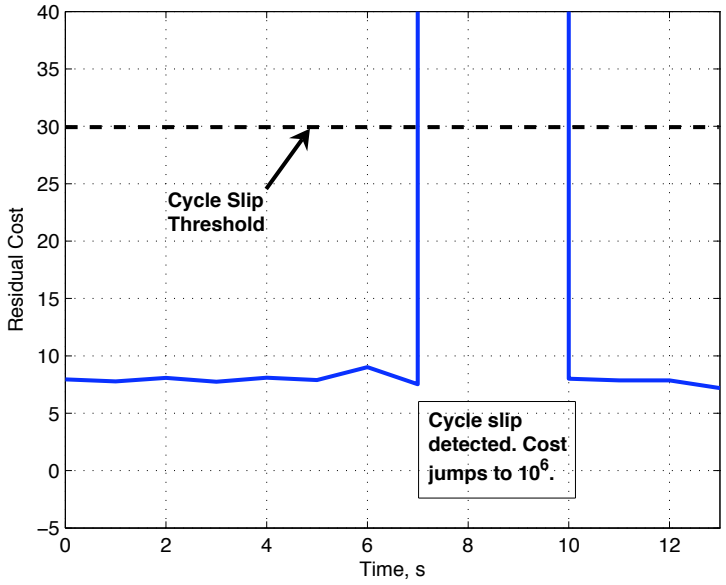

Figure 2.3: Residual cost and cycle slip detection threshold cost.

on which the error occurred. It does this by testing multiple hypotheses. Each hypothesis assumes that a different channel has slipped. To this end, a new ambiguity vector is defined as

$$\mathbf{N}_{new} = \mathbf{N}_{old} + \mathbf{e}^p \Delta n_k^p \tag{2.31}$$

where $p = 1, 2, ..., J$, indicates the assumed slip channel, and $\mathbf{e}^p$ is an $(J-1) \times 1$ vector defined as

$$\mathbf{e}^1 = \begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \mathbf{e}^2 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \mathbf{e}^3 = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \ldots, \mathbf{e}^J = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix} \tag{2.32}$$

The first vector indicates that the cycle slip occurred in the single-differenced measurement that has been subtracted off all of the other single-differenced

measurements as part of the double-differenced data manipulation. Based on the definitions in Eq. (2.11), this is considered to be the $p = 1$ channel of the receiver. For simplicity, the measurement blocks defined in Eq. (2.23) are redefined in more compact notation here.

$$\tilde{H}_{mx_k} = \begin{bmatrix} \tilde{H}_{x_{P_k}} \\ \tilde{H}_{x_{\phi_k}} \end{bmatrix}; \qquad \tilde{H}_{mN_k} = \begin{bmatrix} 0 \\ \tilde{H}_{N_{\phi_k}} \end{bmatrix} \tag{2.33}$$

The measurement block associated with the position correction estimate R_{mx_k} is QR-factorized

$$Q_{x_k} \begin{bmatrix} R_{xx_k} \\ 0 \end{bmatrix} = \tilde{H}_{mx_k} \tag{2.34}$$

and the problem is transformed

$$\begin{bmatrix} R_{xN_k} \\ R_{aN_k} \end{bmatrix} = Q_{x_k}^T \tilde{H}_{mN_k} \tag{2.35}$$

$$\begin{bmatrix} \mathbf{z}_{x_k} \\ \mathbf{z}_{a_k} \end{bmatrix} = Q_{x_k}^T \begin{bmatrix} \tilde{\mathbf{z}}_{P_k} \\ \tilde{\mathbf{z}}_{\phi_k} \end{bmatrix} \tag{2.36}$$

The problem is now ready for sequential hypothesis testing. If there were no cycle slips, then the information equations for $\mathbf{N}$ would take the form

$$\begin{bmatrix} R_{aN_k} \\ \hat{R}_{NN_{k-1}} \end{bmatrix} \mathbf{N} = \begin{bmatrix} \mathbf{z}_{a_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} \tag{2.37}$$

The p^{th} cycle slip hypothesis is that the first row of this system should be modified to be $R_{aN_k}[\hat{\mathbf{N}}_k^p + \mathbf{e}^p \Delta\hat{n}_k^p] = \mathbf{z}_{a_k}$, while the second row remains $\hat{R}_{NN_{k-1}} \hat{\mathbf{N}}_k^p = \hat{\mathbf{z}}_{N_{k-1}}$, where $\hat{\mathbf{N}}_k^p$ should be $\mathbf{N}_{old}$, and where $\Delta\hat{n}_k^p$ should be the slip magnitude. Rewritten in block form, these equations become

$$\begin{bmatrix} R_{aN_k}\mathbf{e}^p & R_{aN_k} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} \begin{bmatrix} \Delta\hat{n}_k^p \\ \hat{\mathbf{N}}_k^p \end{bmatrix} = \begin{bmatrix} \mathbf{z}_{a_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} \tag{2.38}$$

QR-factorization is applied to the augmented matrix to isolate the blocks associated with $\Delta\hat{n}_k^p$ and $\hat{\mathbf{N}}_k^p$

$$Q_{N_k}^p \begin{bmatrix} R_{\Delta n \Delta n_k}^p & R_{\Delta n N_k}^p \\ 0 & \hat{R}_{NN_k}^p \end{bmatrix} = \begin{bmatrix} R_{aN_k}\mathbf{e}^p & R_{aN_k} \\ 0 & \hat{R}_{NN_{k-1}} \end{bmatrix} \tag{2.39}$$

and again the non-homogeneous term is also transformed

$$\begin{bmatrix} z_{\Delta n_k}^p \\ \hat{\mathbf{z}}_{N_k}^p \end{bmatrix} = (Q_{N_k}^p)^T \begin{bmatrix} \mathbf{z}_{a_k} \\ \hat{\mathbf{z}}_{N_{k-1}} \end{bmatrix} \tag{2.40}$$

The transformed quantities $\hat{R}_{NN_k}^p$ from the second row of Eq. (2.39) and $\hat{\mathbf{z}}_{N_k}^p$ from the second row of Eq. (2.40) are then passed to the LAMBDA integer estimation algorithm, which minimizes the following cost function:

$$J(\hat{\mathbf{N}}_k^p) = \frac{1}{2}(\hat{R}_{NN_k}^p \hat{\mathbf{N}}_k^p - \hat{\mathbf{z}}_{N_k}^p)^T (\hat{R}_{NN_k}^p \hat{\mathbf{N}}_k^p - \hat{\mathbf{z}}_{N_k}^p) \tag{2.41}$$

Equations (2.38) through (2.41) are evaluated for $p = 1, 2, ..., J$. The minimum-cost solution indicates the cycle slip channel.

The next step is to calculate the number of cycles slipped under the assumption that the receiver may temporarily slip a non-integer number of cycles. The real-valued cycle slip variable $\Delta\hat{n}_k^p$ is calculated using the remaining blocks from Eqs. (2.39) and (2.40), setting $\hat{\mathbf{N}}_{opt_k} = \hat{\mathbf{N}}_{opt_k}^p$ for the p value associated with the correct slip channel assumption. At this point, all the p superscripts can be dropped, since the algorithm will use the minimum-cost p value.

$$\Delta\hat{n}_k = R_{\Delta n \Delta n_k}^{-1}[z_{\Delta n_k} - R_{\Delta n N_k}\hat{\mathbf{N}}_{opt_k}] \tag{2.42}$$

The position corrections are calculated using $\Delta\hat{n}_k$, $\hat{\mathbf{N}}_{opt_k}$, and the appropriate quantities from Eqs. (2.34) through (2.36)

$$\mathbf{x}_{opt_k} = R_{xx_k}^{-1}[\mathbf{z}_{x_k} - R_{xN_k}(\hat{\mathbf{N}}_{opt_k} + \mathbf{e}^p \Delta\hat{n}_k)] \tag{2.43}$$

The CSR algorithm then assembles a larger cost function

$$J(\Delta n_k, \mathbf{N}_k) = \frac{1}{2}\left(\begin{bmatrix} R_{\Delta n \Delta n_k} & R_{\Delta n N_k} \\ 0 & \hat{R}_{NN_k} \end{bmatrix}\begin{bmatrix} \Delta n_k \\ \mathbf{N}_k \end{bmatrix} - \begin{bmatrix} z_{\Delta n_k} \\ \mathbf{z}_{N_k} \end{bmatrix}\right)^T \times \left(\begin{bmatrix} R_{\Delta n \Delta n_k} & R_{\Delta n N_k} \\ 0 & \hat{R}_{NN_k} \end{bmatrix}\begin{bmatrix} \Delta n_k \\ \mathbf{N}_k \end{bmatrix} - \begin{bmatrix} z_{\Delta n_k} \\ \mathbf{z}_{N_k} \end{bmatrix}\right) \tag{2.44}$$

where Δn_k is a cycle slip variable that is constrained to be an integer, and where $\mathbf{N}_k$ is a time-varying integer ambiguity vector. This problem is solved using the LAMBDA method. Continuing under the assumption that the number of cycles slipped may not be an integer number of cycles during a multiple time step disruption, these solutions from LAMBDA are not used for the navigation solution, but are saved for comparison in the subsequent time steps.

The algorithm passes $\hat{R}_{NN_k}$ and $\hat{\mathbf{z}}_{N_k}$ forward to the next time step as a priori information, and repeats the CSR steps outlined above. This time, however, it uses the cycle slip channel estimate from the previous time step instead of searching through all the channel hypotheses. The stopping criterion for the CSR algorithm is that the integer cycle slip determined by minimizing the cost function in Eq. (2.44) remains unchanged for one time step, i.e. $\Delta n_{opt_k} = \Delta n_{opt_{k-1}}$, and that the integer ambiguity vector estimate from time k equals the ambiguity vector estimate from just before the cycle slip, i.e. $\mathbf{N}_{opt_k} = \mathbf{N}_{old}$.

At this point, the CSR algorithm must perform some manipulations in order to pass the appropriate a priori information to the executive estimator. It recognizes that the new a priori square-root information equation for the ambiguities should be equivalent to

$$\begin{bmatrix} R_{\Delta n N_k} \\ R_{NN_k} \end{bmatrix}\mathbf{N}_{old} = \begin{bmatrix} R_{\Delta n N_k} \\ R_{NN_k} \end{bmatrix}\{\mathbf{N}_{new} - \mathbf{e}^p \Delta n_{opt_k}\} = \begin{bmatrix} z_{\Delta n_k} - R_{\Delta n \Delta n_k}\Delta n_{opt_k} \\ \hat{\mathbf{z}}_{N_k} \end{bmatrix} - \begin{bmatrix} v_{\Delta n_k} \\ \hat{v}_{N_k} \end{bmatrix} \tag{2.45}$$

It QR-factorizes the coefficient matrix of $\mathbf{N}_{new}$

$$\bar{Q}_{N_k}\begin{bmatrix}\bar{R}_{NN_k}\\ 0\end{bmatrix}=\begin{bmatrix}R_{\Delta nN_k}\\ \hat{R}_{NN_k}\end{bmatrix} \tag{2.46}$$

and collects and transforms the equation's non-homogeneous terms

$$\begin{bmatrix}\bar{\mathbf{z}}_{N_k}\\ \bar{\mathbf{z}}_{r_k}\end{bmatrix}=\bar{Q}_{N_k}^{T}\begin{bmatrix}\{z_{\Delta n_k}-R_{\Delta n\Delta n_k}\Delta n_{opt_k}+R_{\Delta nN_k}\mathbf{e}^{p}\Delta n_{opt_k}\}\\ \{\mathbf{z}_{N_k}+R_{NN_k}\mathbf{e}^{p}\Delta n_{opt_k}\}\end{bmatrix} \tag{2.47}$$

It then sets $\hat{R}_{NN_{k-1}} = \bar{R}_{NN_k}$ and $\hat{\mathbf{z}}_{N_{k-1}} = \bar{\mathbf{z}}_{N_k}$, and returns to the original solution algorithm.

The cycle slip recovery algorithm is illustrated in block diagram form in Fig. 2.4 and is summarized in the following steps:

1. Save the integer ambiguity estimate calculated just before the cycle slip, i.e. $\mathbf{N}_{old} = \mathbf{N}_{opt}$.
2. Perform hypothesis testing by optimizing the cost in Eq. (2.41) for each channel. Let the slip channel be called channel p.
3. Calculate/retain the integer ambiguity estimate $\hat{\mathbf{N}}_k^p$ associated with the slip channel.
4. Calculate the real-valued number of cycles that have slipped $\Delta\hat{n}_k^p$ using Eq. (2.42) and calculate the position corrections $\mathbf{x}_k$ using Eq. (2.43).
5. Use the LAMBDA method to determine the minimizing integer values Δn_k and $\mathbf{N}_k$ associated with the cost in Eq. (2.44).
6. Set $k = k + 1$, retrieve new measurements, perform transformations in Eqs. (2.34) through (2.36), repeat Steps 3, 4, and 5, then continue to Step 7.

7. Check if $\Delta n_{opt_k} = \Delta n_{opt_{k-1}}$ and if $\mathbf{N}_{opt_k} = \mathbf{N}_{old}$. If both conditions are satisfied, go to Step 8. If either condition is not satisfied, set $k = k + 1$, retrieve new measurements, perform transformations in Eqs. (2.34) through (2.36), and repeat Steps 3, 4, 5, and 7.

8. Calculate the appropriate a priori information using Eqs. (2.46) and (2.47), and return to Step 2 of the executive solution algorithm.

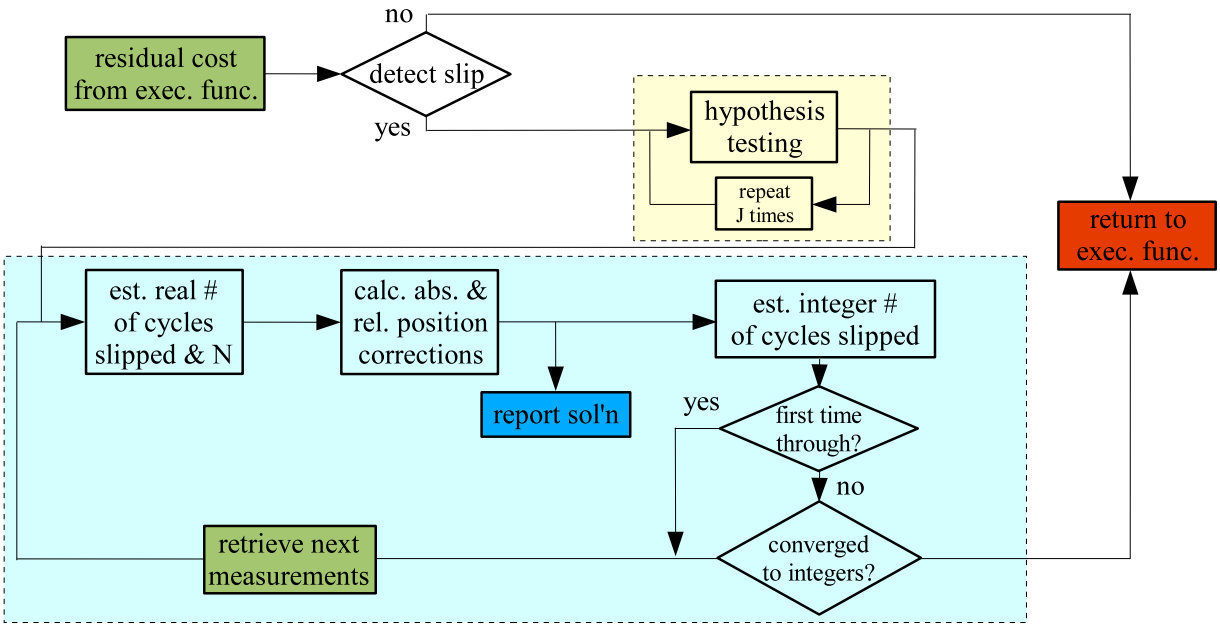

Figure 2.4: Cycle slip recovery algorithm.

2.7 Results from Hardware Receiver/Simulator Data and Off-Line MATLAB Simulator Data

Evaluation of a spacecraft relative navigation estimator requires the use of a sophisticated simulations and space-qualified GPS receivers. A state-of-the-art GPS simulation laboratory exists at NASA Goddard Space Flight Center

in the Formation Flying Testbed (FFTB). This laboratory includes a multichannel satellite navigation simulator capable of replicating the GPS RF signals that would be received by antennas onboard orbiting platforms.[25] A receiver connected to the simulator behaves as if it were onboard an actual spacecraft (except for the environmental effects such as radiation and temperature extremes). A commercial-off-the-shelf (COTS) GPS receiver will not navigate under these conditions due to government-imposed altitude and speed restrictions. Dedicated space-qualified receivers must be used instead, and such receivers, designed by the German space agency DLR, are available at the FFTB. The Orion receivers were designed specifically for use in satellite formation flying. They were built using a COTS chipset and software package, but they include significant software modifications to ensure that they measure and report carrier-phase data correctly.[16]

The signal simulator and the DLR Orion receivers have been used to replicate on-orbit relative navigation CDGPS measurements. The data have simulated a 2-spacecraft, 1-km-baseline formation in LEO. Ionosphere and multipath errors have not been simulated in order to obtain a measure of the relative navigation estimator's baseline performance under idealized conditions. These results have been used to compare the new algorithm's performance in LEO scenarios to previous work. Using these data, the algorithm typically converges to the correct integer ambiguities in 1-2 one-second time steps and reaches a probability that the estimated integers are the correct integers of 0.99 in 3-4 time steps when tracking 8 or more GPS satellites. The probability calculation is based on the Bayesian analysis method presented in Ref. 22. It calculates the probability that the integer estimates are correct conditioned on the data and on the a priori

ambiguity information according to the following formula:

$$P(\mathbf{N}_{opt}|\hat{\mathbf{z}}_N) = \frac{exp[-J(\mathbf{N}_{opt})]}{\sum_{m=1}^{M} exp[-J(\mathbf{N}_m)]} \tag{2.48}$$

where $\mathbf{N}_m$ for $m = 1, 2, \ldots, M$ constitute the set of all remotely possible integer vectors determined by the LAMBDA/ILLS algorithm. The size of this set and, correspondingly, the upper limit M of the summation in the denominator of Eq. (2.48) are chosen to be large enough that the summation approximates the infinite sum required by the Bayesian analysis.

A comparison between the relative position estimates and the true relative positions that are available from the signal simulator has found 3-dimensional relative position errors of 3 mm. Figure 2.5 shows about 200 s of the estimated baseline distance and the true baseline distance for a typical test case. In this case, and many others, the correct integer ambiguities were estimated in the first time step. Figure 2.6 plots the position error magnitude time history of the same sample period. These accuracies, achieved using L1-only measurements, are similar to those achieved by previous dual-frequency estimators that incorporate LAMBDA and the convergence times are faster,[11,12] although a direct comparison is not valid given the idealized conditions of the simulation.

The estimator has been tested under GEO and HEO scenarios using data that has been generated by an off-line MATLAB truth-model simulation. This simulator is described in detail in Ref. 23 and includes realistic error models. The effects of the ionosphere are simulated using a modified version of the standard GPS broadcast model that incorporates altitude dependence into its electron density model. The total electron content for each channel is calculated by numerically integrating along the LOS vectors. The multipath error is modeled as being a function of the LOS directions to each GPS satellite as mea-

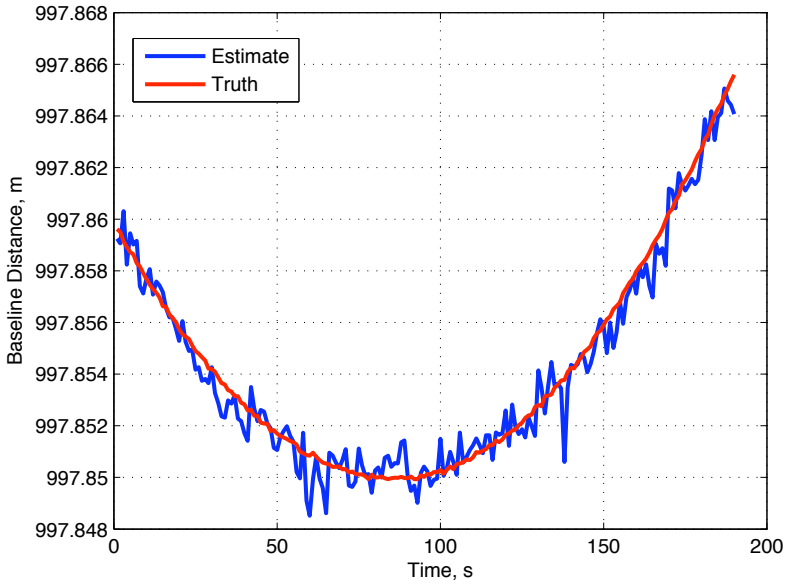

Figure 2.5: 1-*km*-baseline, LEO scenario. Estimated and true baseline distance.

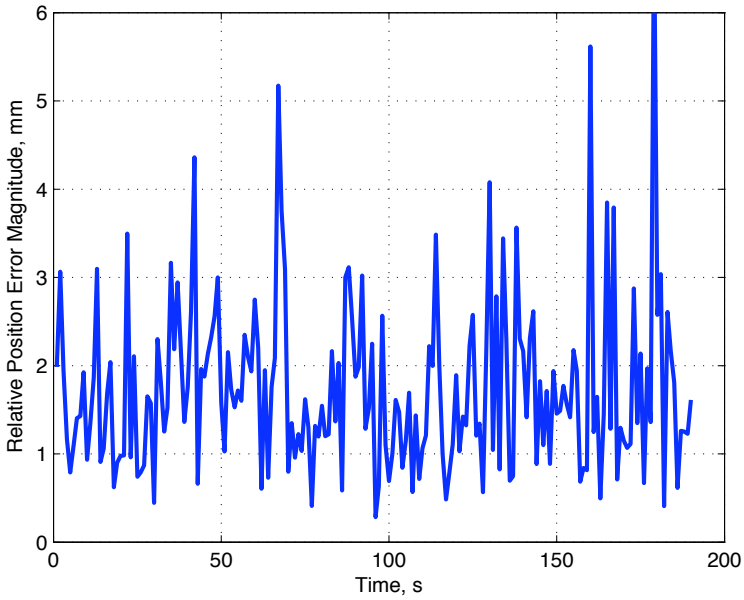

Figure 2.6: 1-*km* baseline, LEO scenario. Time history of the relative position error magnitude.

sured with respect to each user spacecraft's body coordinates. This function is implemented using a spherical harmonic expansion, allowing the model to capture the local azimuth and elevation dependencies of the multipath errors. The model assumes that the user spacecraft are relatively small and simple, having no large appendages. It would not be valid for large, complex spacecraft, e.g. the International Space Station. The maximum carrier-phase multipath error is 5 *mm*. These detailed error models allow the simulator to replicate the expected real-world conditions a receiver would experience at high altitudes.

A representative GEO scenario consists of a 2-spacecraft formation with a 3 km baseline. It is assumed that the receivers include the weak signal acquisition and tracking algorithms described in Ref. 20 and Ref. 21 that allow the use of the very weak side lobes of the GPS satellites' broadcast beams. Under the assumption that signals can be tracked down to C/N_0 = 18 dB-Hz, receivers in GEO can track up to 11 GPS signals with a GDOP of between 5 and 13. For scenarios where 8 or more GPS satellites are tracked and where the GDOP is less than 12, the algorithm converges to the correct integer ambiguities in 1-2 thirty-second time steps and reaches 99 percent certainty in 3-4 time steps. It produces relative position error magnitudes of less than 0.1 m. Figure 2.7 shows the time histories of the position error components for this case.

A representative HEO scenario places the spacecraft in a 1.2 × 18 Re orbit with a baseline of 10 km at apogee. In this scenario, it is assumed that the receivers can track signals down to C/N_0 = 12 dB-Hz. This threshold allows each receiver to track up to 9 satellites at apogee with a GDOP between 110 and 300. In a scenarios where 7 or more GPS satellites are tracked and where the GDOP is less than 250, the algorithm converges to the correct integer ambigui-

ties in 6-7 thirty-second time steps and reaches 99 percent certainty in 9-10 time steps. It produces relative position errors of less than 3 m. Figure 2.8 shows the time histories of the position error components for this case. Notice that the most significant errors occur in the weakly observable radial direction. Errors in this direction might be reducible by the addition of spacecraft relative dynamics to the solution algorithm through filtering, although this improvement would come at the cost of added complexity.

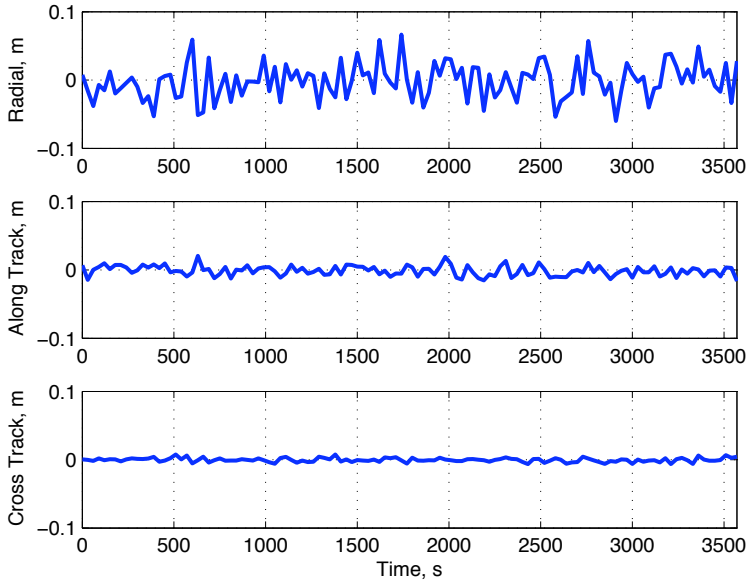

Figure 2.7: Time histories of the relative position error components for a geostationary formation.

The hardware-in-the-loop data sets collected with the Orion receivers and the FFTB simulators included many carrier-phase jumps and offered an opportunity to test the cycle slip detection and recovery system. Given that the estimator typically converges to the correct integer ambiguities in 1-2 one-second time steps with no a priori information, a reasonable alternative approach to this paper's cycle slip recovery system would be to set the uncertainties associated with the ambiguity estimates to very large numbers after a cycle slip is

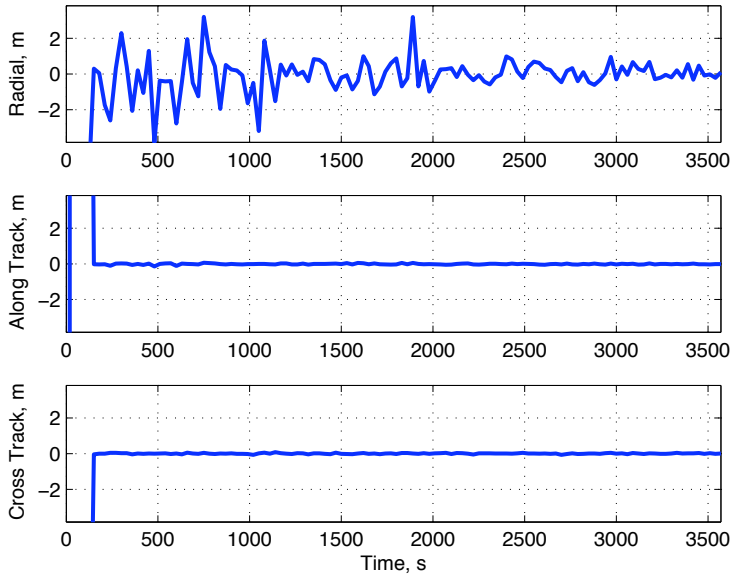

Figure 2.8: Time histories of the relative position error components for a formation in a 1.2 × 18 *Re* orbit when at a radial distance of 17.8 *Re*.

detected, essentially resetting the estimator. This ad hoc method produces a jump in the relative position error on the order of several meters, as illustrated in Fig. 2.9. The new CSR algorithm improves the estimator's transient performance during slips. Figure 2.10 shows the error magnitude time history for the same block of data shown in Fig. 2.9, but for an estimator that uses the new slip recovery algorithm. The cycle slip detection and recovery system correctly detects a measurement error and correctly identifies the channel on which the slip occurred. The estimate of the number of cycles slipped converges to an integer in 3 one-second time steps. During convergence, the relative navigation solution, calculated with the real-valued cycle slip variable, shows no significant accuracy degradation.

Note that the cycle slip detection and recovery algorithm has been tested only in LEO. It is expected to show similarly good performance in GEO because

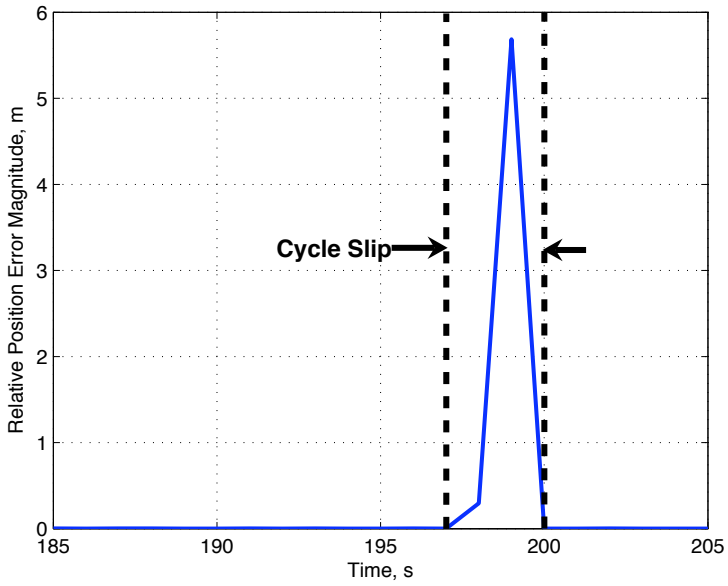

Figure 2.9: Time history of the relative position error magnitude during a cycle slip event when using an ad hoc approach that discards all a priori ambiguity information at the onset of the slip.

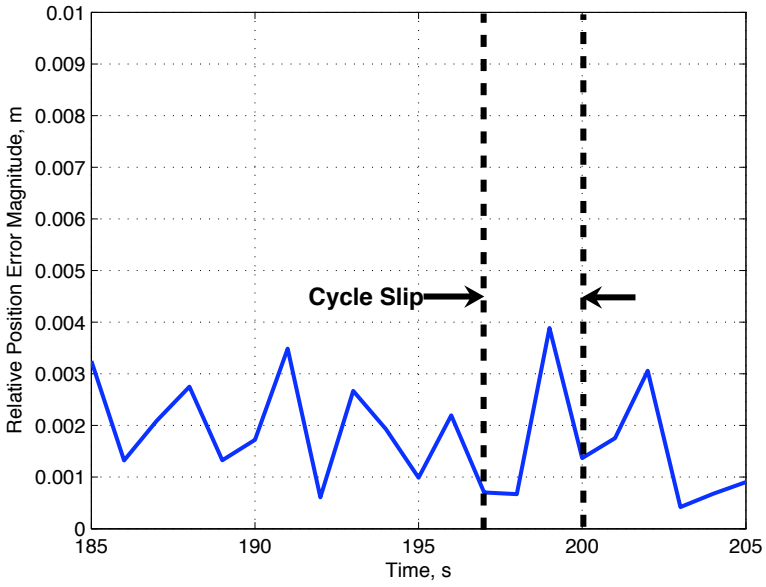

Figure 2.10: Time history of the relative navigation error magnitude during a cycle slip event when using the new detection and recovery algorithm.

of the demonstrated ability to rapidly resolve ambiguities at those altitudes, even without a priori information. Performance in HEO may be less robust because ambiguity resolution takes longer at those altitudes.

2.8 Conclusions

A new satellite relative navigation estimator has been developed. It incorporates integer ambiguity resolution in an optimal manner. The algorithm, which uses an improved carrier-phase measurement model, reduces the measurement ambiguity search volume by using both pseudorange and carrier-phase measurements in the calculation of the solution and by imposing an integer condition on the double-differenced carrier-phase measurement ambiguities. The reduced search volume promotes rapid convergence to the correct ambiguities. The algorithm utilizes an efficient decorrelation and integer search scheme to rapidly calculate the correct ambiguities, and it employs a robust cycle slip detection and recovery system.

The new estimator can converge to the correct integer ambiguities in 1-2 one-second measurement steps in a 1-km-baseline LEO formation and can reach an ambiguity validation certainty of 99 percent in 3-4 measurement steps as determined by idealized hardware-in-the-loop tests. In off-line GEO simulations of a 3-km-baseline formation, the algorithm can converge to the correct integers in 1-2 thirty-second time steps and can reach an ambiguity validation certainty of 99 percent in 3-4 time steps. In off-line HEO simulations at a radial distance of 17.8 Re, the estimator can converge to the correct integers in about 3 minutes and can reach a certainty of 99 percent in 9-10 thirty-second time steps. The

convergence rates in GEO and HEO indicate that, with careful implementation, LAMBDA techniques can allow CDGPS techniques to be used in high-altitude formations with L1-only measurements.

The estimator achieves 3-dimensional relative position errors on the order of 3 mm over a 1 km baseline in LEO, 0.1 m over a 3 km baseline in GEO, and 3 m over a 10 km baseline at apogee in HEO. These relative navigation accuracies exceed those required for many formation flying missions. The results, however, rest on the assumption that error models in the off-line simulator are accurate. These models should be validated against real data when high-altitude, weak-signal data become available. Further improvement may be achieved by including dynamics models in the estimator, thus making it more robust to random errors.

The cycle slip detection and recovery system correctly detects carrier-phase measurement errors, identifies the slip channel, estimates the number of cycles slipped, and returns the proper a priori information to the executive estimator. It does so without degrading the relative navigation solution during the slip transient.

Acknowledgments

This work has been supported in part by a Graduate Student Researcher Program Fellowship administered through NASA Goddard Space Flight Center with Russell Carpenter acting as Technical Advisor and in part by NASA Cooperative Agreement No. NCC5-722, which has been monitored by Michael C. Moreau. The authors would like to thank Dr. Moreau for facilitating the use

of NASA's Formation Flying Testbed and Oliver Montenbruck of DLR for developing very capable, space-qualified receivers and for making his hardware-simulator data available to us.

CHAPTER 3

CARRIER-PHASE DIFFERENTIAL GLOBAL POSITION SYSTEM KALMAN FILTER FOR HIGH-ALTITUDE SPACECRAFT NAVIGATION

S. Mohiuddin and M.L. Psiaki, "Carrier-Phase Differential Global Position System Kalman Filter for High-Altitude Spacecraft Navigation," *Journal of Guidance, Control, and Dynamics*, Vol. 31, No. 4, July-August 2008.

3.1 Abstract

A new navigation Kalman filter has been developed that uses carrier-phase differential Global Positioning System techniques in a framework that includes integer ambiguities to estimate the relative states of high-altitutde formation-flying spacecraft. This model-based approach to relative navigation allows spacecraft formations to use carrier-phase techniques above the Global Positioning System constellation. The filter uses dynamics models for the spacecraft orbits, receiver clocks, ionospheric total electron content, and Global Positioning System satellite residual position and clock errors. The process noise driving the orbital dynamics is separated into a common-mode part and lower intensity differential-mode part in order to better model how disturbances influence formations. The ionospheric total electron content model contains information that allows the filter to decide when to use the dual-frequency measurements to correct for the ionosphere and when to use them to aid in integer ambiguity resolution. The filter tracks un-differenced carrier-phase ambiguities, but still resolves the double-differenced integer ambiguities for use in the relative navigation solution. Monte Carlo simulations are used to evaluate the filter's per-

formance. In geostationary scenarios, the filter's mean relative position error magnitude is 1.2 cm, and its maximum error magnitude is 6 cm. In high-Earth orbit scenarios at an apogee distance of 18 Earth radii, the filter's mean error magnitude is 11 *cm*, and its maximum error magnitude is 54 cm. The correct integer ambiguities are resolved in one or two 30-second measurement steps in most cases.

3.2 Introduction

Formation flying of spacecraft is a key component in many space mission concepts. Formation-based sensing, for certain types of missions, can be more cost effective, redundant, flexible, and evolvable than single platform sensing that yields similar performance. Formation flying, however, is technically challenging, requiring each satellite to perform autonomous and accurate relative navigation. Carrier-phase differential Global Positioning System (CDGPS) techniques offer the required level of autonomy and accuracy, especially when combined with dynamics models through filtering. CDGPS techniques take the precise, but biased, GPS carrier-phase measurements, resolve the biases through integer estimation techniques, and retain accurate information about the relative states of a pair of receivers. These techniques have already been demonstrated for low Earth orbit (LEO) spacecraft relative positioning in simulations[4,11,13,16] and in off-line processing of flight data,[12] achieving sub-cm level accuracy. This project applies CDGPS techniques to formations flying in geostationary orbits (GEO) and high-altitude Earth orbits (HEO), both of which require GPS receivers to operate above the GPS constellation. The orbits under consideration in this study are illustrated in Fig. 3.1.

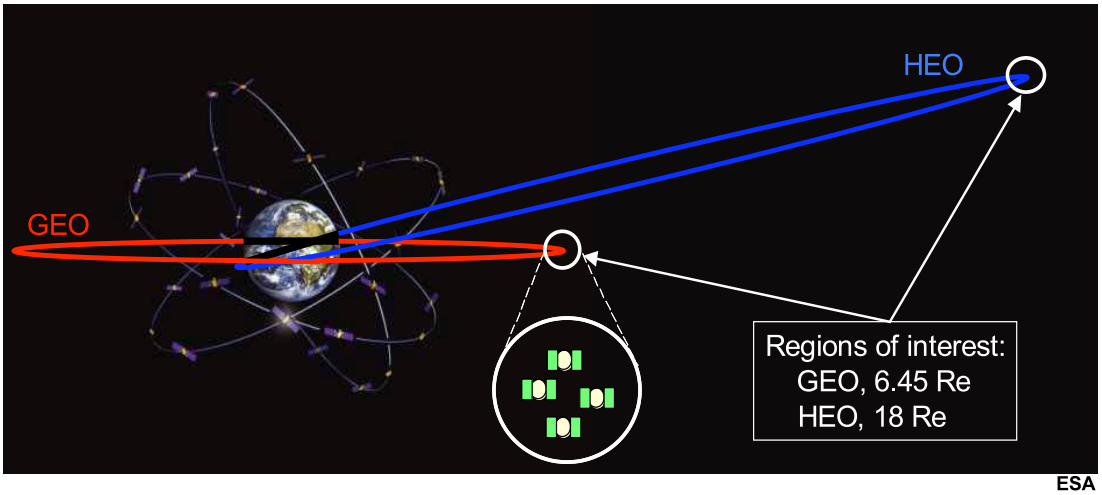

Figure 3.1: Relative navigation scenarios.

A navigation system designed to work in high-altitude orbits must overcome several significant challenges. First, since the GPS receivers will be operating outside of the main broadcast beams of most of the GPS satellites, they must acquire and track the weak side lobes of the broadcast signals. It is assumed in this study that weak signal GPS receivers that use the techniques in Refs. 19 and 21 are available. Second, the integer nature of the double-differenced carrier-phase measurement ambiguities can be corrupted if significant amounts of ionospheric total electron content (TEC) remain after the CDGPS differencing operations for GPS signals whose ray paths pass near the Earth. Differential TEC must be estimated, and its effects must be removed in order to resolve the ambiguities as exact integers, a necessary step for precise relative navigation.

Third, poor geometric dilution of precision (GDOP) and slow line-of-sight (LOS) vector dynamics between the receivers and the GPS satellites make the integer ambiguities weakly observable, hindering the estimator's ability to resolve the values on the fly. Powerful integer-constrained linear least-squares estimation techniques, typically called LAMBDA/integer-linear-least-squares techniques (LAMBDA/ILLS), must be optimally integrated into the solution algorithm in order to overcome this challenge.

Although the high-altitude environment is challenging, it offers at least one advantage that is unavailable closer to Earth. The GPS receivers in high-altitude orbits will operate almost entirely outside of the ionosphere. Since the GPS satellites also operate outside of the ionosphere, many cases arise in which a broadcast signal's ray path never passes through a dense portion of the ionosphere. The effects of the differential TEC on that signal may be neglected. If the receiver-Earth-transmitter relative geometry is considered, such cases may be predicted, eliminating the need for the estimator use the dual-frequency measurements to remove the effects of the ionosphere. The estimator is thus free to use those measurements for other important purposes, as will be discussed below. A realistic model of the magnitude and evolution of the TEC effects, therefore, could result in a more flexible estimator that makes multi-modal use of the dual-frequency measurements.

Most of the previous research related to using CDGPS techniques for the relative navigation of satellite formations has focused on LEO applications. Researchers have used a variety of techniques, including single- and dual-frequency pointwise estimators[11,16] and single- and dual-frequency extended Kalman filters,[4,12,13] in order to achieve cm- or sub-cm level relative position

accuracy in simulations and in post-processing of flight data. These algorithms benefit from the good signal environment and fast LOS vector dynamics found at low altitudes and cannot be directly applied to high-altutitude formation scenarios. Other researchers have extended the use of CDGPS techniques to high altitudes,[15,24] but only in the context of pointwise estimation. These high-altutide pointwise estimators, however, suffer relatively large errors in the weakly observable radial direction, they do not produce velocity estimates, and they do not estimate TEC, or they use simplistic estimation techniques that include little a priori TEC knowledge.

The goal of this paper is to improve high-altitude relative navigation performance by using dual-frequency techniques and Kalman filtering techniques that exploit dynamics models. This research approaches this task in two steps. First, the single-frequency pointwise estimation techniques used in Ref. 15 and the GPS satellite residual position and clock error estimation techniques used in Ref. 24 are integrated into an extended Kalman filter. The filter augments its states to include spacecraft velocities and receiver clock rates. The coupling of the radial and along-track directions through the orbital dynamics and the inclusion of receiver clock dynamics models improves the relative position accuracy, especially in the radial direction, which is the direction with the largest errors.[11,15] The second step is to include dual-frequency measurements and a realistic dynamics model for the TEC, with special attention given to identifying those situations in which the differential TEC is significant and those in which it can be neglected. This model gives the filter the information it needs to decide whether to use the dual-frequency measurements to estimate and remove the effects of the ionosphere or to aid in integer ambiguity resolution through implicit wide laning.

The remainder of the paper is divided into 5 major sections. Section II describes the filter's dynamics models, Section III presents the carrier-phase and pseudorange measurement models, Section IV defines the Kalman filter algorithm, Section V evaluates the filter's performance using truth-model simulation results, and Section VI gives the conclusions.

3.3 Dynamics Modeling

Three different relative navigation estimators — an L1-L2 filter, an L1-only filter, and an L1-only pointwise estimator — have been developed and are compared later in this paper.* All three have been derived using similar techniques, but only the derivation of the L1-L2 filter is presented here. Where appropriate, the differences among these estimators are pointed out. The L1-L2 relative navigation filter is implemented in two steps, the propagation step and the measurement update step. The dynamics models for the propagation step are presented in this section, and the measurement equations for the measurement update step are presented in the next section. This paper focuses on the relative navigation of a pair of satellites, although the techniques are applicable to formations of an arbitrary number of spacecraft.

The propagation step is based on dynamics models for the orbits of the individual spacecraft, for the independent receiver clocks, for the GPS satellite residual position and clock errors, for the TEC along the LOS vectors from each receiver to each GPS satellite, and for the carrier-phase measurement ambigu-

*In this paper, "L1" refers to standard civilian GPS signal broadcast at 1575.42 MHz, and "L2" refers to the new civilian GPS signal broadcast at 1227.60 MHz that is being added to the system as part of GPS modernization.

ities. The pointwise estimator includes limited dynamics models: a Markov model for the net LOS effect of the GPS spacecraft residual position and clock errors and a constant model for the ambiguities. The L1 filter includes the same models as the L1-L2 filter, but does not model or estimate TEC.

The L1-L2 relative navigation filter uses absolute orbital dynamics models for each spacecraft. These models include a 2×2 gravity model and Sun and moon disturbances. The nonlinear, continuous time equations that describe the motion of an Earth orbiting satellite are expressed in six states for each spacecraft with additive process noise.

$$\begin{bmatrix} \dot{\mathbf{X}}_{sc_A}(t) \\ \dot{\mathbf{X}}_{sc_B}(t) \end{bmatrix} = \begin{bmatrix} \mathbf{F}[\mathbf{X}_{sc_A}(t), t] \\ \mathbf{F}[\mathbf{X}_{sc_B}(t), t] \end{bmatrix} + D(t)\mathbf{w}_{sc}(t) \tag{3.1}$$

The state vector contains the position and velocity of each satellite expressed in Earth centered inertial (ECI) coordinates, and the function F includes the nonlinear accelerations of the satellite. The following equations are presented for user spacecraft A, but the same equations apply for spacecraft B.

$$\mathbf{X}_{sc_A}(t) = \begin{bmatrix} x_A(t) \\ y_A(t) \\ z_A(t) \\ \dot{x}_A(t) \\ \dot{y}_A(t) \\ \dot{z}_A(t) \end{bmatrix}; \qquad \mathbf{F}[\mathbf{X}_{sc_A}(t), t] = \begin{bmatrix} \dot{x}_A(t) \\ \dot{y}_A(t) \\ \dot{z}_A(t) \\ g_x[\mathbf{X}_{sc_A}(t), t] \\ g_y[\mathbf{X}_{sc_A}(t), t] \\ g_z[\mathbf{X}_{sc_A}(t), t] \end{bmatrix} \tag{3.2}$$

where the functions g_x, g_y, and g_z are gravitational accelerations that include Sun and moon effects.

The spacecraft state process noise vector $\mathbf{w}_{sc}$ in Eq. (3.1) is modeled as a zero mean, Gaussian perturbation to the spacecraft accelerations. The noise is di-

vided into two parts, one that is common to both satellites and one that affects each satellite differentially. The relative proximity of the formation satellites in the much larger orbit indicates that each satellite's motion is governed mostly by the same influences. Only a small portion of the perturbations to the nominal dynamics affects the satellites differentially. By separating the process noise into a larger common-mode noise and a smaller differential-mode noise, the physical situation of relative navigation is better modeled. The process noise vector and its coefficient matrix used to achieve the separation are

$$\mathbf{w}_{sc}(t) = \begin{bmatrix} \mathbf{w}_{AB}(t) \\ \delta\mathbf{w}_{AB}(t) \end{bmatrix}; \qquad D(t) = \begin{bmatrix} 0 & 0 \\ I & I \\ 0 & 0 \\ I & -I \end{bmatrix} = \begin{bmatrix} D_A \\ D_B \end{bmatrix} \tag{3.3}$$

where $\mathbf{w}_{AB}(t)$ is the common-mode process noise and $\delta\mathbf{w}_{AB}(t)$ is the much lower intensity differential-mode process noise. The process noise acceleration vectors for each spacecraft are thus defined as $\mathbf{w}_A(t) = \mathbf{w}_{AB}(t) + \delta\mathbf{w}_{AB}(t)$ and $\mathbf{w}_B(t) = \mathbf{w}_{AB}(t) - \delta\mathbf{w}_{AB}(t)$.

The discrete time version of the orbital dynamics model takes the form

$$\mathbf{x}_{sc_k} = \begin{bmatrix} \mathbf{X}_{sc_A}(t_k) \\ \mathbf{X}_{sc_B}(t_k) \end{bmatrix} = \mathbf{f}_{sc}\left(\begin{bmatrix} \mathbf{X}_{sc_A}(t_{k-1}) \\ \mathbf{X}_{sc_B}(t_{k-1}) \end{bmatrix}, \mathbf{w}_{sc_{k-1}}, t_{k-1} \right) = \mathbf{f}_{sc}(\mathbf{x}_{sc_{k-1}}, \mathbf{w}_{sc_{k-1}}, t_{k-1}) \tag{3.4}$$

where this equation also serves to define the dual-spacecraft discrete time state vector $\mathbf{x}_{sc_k}$ at sample time t_k. The discrete time nonlinear dynamics function $\mathbf{f}_{sc}$ that is used in Eq. (3.4) is evaluated by integrating the continuous time dynamics in Eq. (3.1) from time t_{k-1} to time t_k using the zero-order hold assumption on the process noise.

$$\begin{bmatrix} \mathbf{w}_{AB}(t) \\ \delta\mathbf{w}_{AB}(t) \end{bmatrix} = \mathbf{w}_{sc_{k-1}} \qquad for \quad t_{k-1} \le t < t_k \tag{3.5}$$

The linearized version of Eq. (3.4) takes the form

$$\mathbf{x}_{sc_k} = \mathbf{f}_{sc}(\hat{\mathbf{x}}_{sc_{k-1}}, 0, t_{k-1}) + \Phi_{sc_{k-1}}(\mathbf{x}_{sc_{k-1}} - \hat{\mathbf{x}}_{sc_{k-1}}) + \Gamma_{sc_{k-1}}\mathbf{w}_{sc_{k-1}} \tag{3.6}$$

where $\hat{\mathbf{x}}_{sc_{k-1}}$ is the nominal state at time t_{k-1}. The state transition matrix and the process noise influence matrix associated with Eq. (3.6), both of which are required for the Kalman filter algorithm,[26] take the form

$$\Phi_{sc_{k-1}} = \begin{bmatrix} \Phi_{sc_A}(t_k, t_{k-1}) & 0 \\ 0 & \Phi_{sc_B}(t_k, t_{k-1}) \end{bmatrix}; \qquad \Gamma_{sc_{k-1}} = \begin{bmatrix} \Gamma_{sc_A}(t_k, t_{k-1}) \\ \Gamma_{sc_B}(t_k, t_{k-1}) \end{bmatrix} \tag{3.7}$$

where Φ_{sc_A} and Γ_{sc_A} are determined by integrating continuous time matrix differential equations that are derived from the spacecraft A part of Eq. (3.1), and where Φ_{sc_B} and Γ_{sc_B} are similarly derived from the spacecraft B part of Eq. (3.1). The associated process noise covariance is

$$Q_{sc_{k-1}} = \begin{bmatrix} \frac{\beta_{w_{AB}}}{\Delta t_{k-1}} I & 0 \\ 0 & \frac{\beta_{\delta w_{AB}}}{\Delta t_{k-1}} I \end{bmatrix} \tag{3.8}$$

where I is a 3×3 identity matrix, $\beta_{w_{AB}}$ is the common-mode process noise intensity tuning parameter, and $\beta_{\delta w_{AB}}$ is the much smaller differential-mode tuning parameter.

The receiver clock errors are modeled as evolving according to a two-state random process for each receiver. This model is based on the model presented in Ref. 3. The states are the clock error and the clock error rate. They are both expected to be random walk/drift processes during operation. The following are definitions of the clock model for receiver A, and similar definitions apply for receiver B.

$$\mathbf{x}_{clk_{A_k}} = \begin{bmatrix} c\delta t_{A_k} \\ c\dot{\delta} t_{A_k} \end{bmatrix}; \qquad \Phi_{clk_{A_{k-1}}} = \begin{bmatrix} 1 & \Delta t_{k-1} \\ 0 & 1 \end{bmatrix}; \qquad \mathbf{w}_{clk_{A_{k-1}}} = \begin{bmatrix} w_{\delta t_{A_{k-1}}} \\ w_{\dot{\delta} t_{A_{k-1}}} \end{bmatrix} \tag{3.9}$$

Here, $c\delta t_{A_k}$ is the range equivalent receiver clock correction, $c\dot{\delta} t_{A_k}$ is the range rate equivalent receiver clock drift, Δt_{k-1} is the time step, $w_{\delta t_{A_{k-1}}}$ is the receiver clock drift process noise, and $w_{\dot{\delta} t_{A_{k-1}}}$ is the receiver clock frequency drift process noise. The models for both receiver clocks are combined.

$$\mathbf{x}_{clk_k} = \begin{bmatrix} \mathbf{x}_{clk_{A_k}} \\ \mathbf{x}_{clk_{B_k}} \end{bmatrix}; \qquad \Phi_{clk_{k-1}} = \begin{bmatrix} \Phi_{clk_{A_{k-1}}} & 0 \\ 0 & \Phi_{clk_{B_{k-1}}} \end{bmatrix}; \qquad \mathbf{w}_{clk_{k-1}} = \begin{bmatrix} \mathbf{w}_{clk_{A_{k-1}}} \\ \mathbf{w}_{clk_{B_{k-1}}} \end{bmatrix} \tag{3.10}$$

The clock dynamics mapping for the two-receiver system can be written compactly as

$$\mathbf{x}_{clk_k} = \Phi_{clk_{k-1}} \mathbf{x}_{clk_{k-1}} + \mathbf{w}_{clk_{k-1}} \tag{3.11}$$

The associated process noise covariance is

$$Q_{clk_{AB_{k-1}}} = \begin{bmatrix} S_f \Delta t_{k-1} + \frac{S_g \Delta t_{k-1}^3}{3} & \frac{S_g \Delta t_{k-1}^2}{2} \\ \frac{S_g \Delta t_{k-1}^2}{2} & S_g \Delta t_{k-1} \end{bmatrix}; \qquad S_f = \frac{h_0}{2}; \qquad S_g = 2\pi^2 h_{-2} \tag{3.12}$$

$$Q_{clk_{k-1}} = \begin{bmatrix} Q_{clk_{AB_{k-1}}} & 0 \\ 0 & Q_{clk_{AB_{k-1}}} \end{bmatrix} \tag{3.13}$$

where the covariance is expressed in terms of the typical Allen variance parameters, h_0 and h_{-2}, for a temperature compensated crystal oscillator.

The GPS satellite residual position and clock errors, often referred to as ephemeris errors, are modeled as though they evolve according to a first-order Markov process in order to capture the expected time correlation of these errors. The vector of errors is defined as $\mathbf{e}_k = [e_k^1 \; e_k^2 \; \ldots \; e_k^J]^T$, where e_k^j is the linear combination of the j^{th} GPS satellite's residual position error projected onto the receiver-transmitter LOS vector plus the GPS satellite's distance-equivalent residual clock error. The number of commonly tracked GPS satellites is assumed to be J. The first-order Markov process mapping is

$$\mathbf{e}_k = \alpha_{e_{k-1}} \mathbf{e}_{k-1} + \Gamma_{e_{k-1}} \mathbf{w}_{e_{k-1}} \tag{3.14}$$

where

$$\alpha_{e_{k-1}} = \exp[-\Delta t_{k-1}/\tau_e] \tag{3.15}$$

$$\Gamma_{e_{k-1}} = \left(\sigma_e \sqrt{1 - \exp[-2\Delta t_{k-1}/\tau_e]}\right) I \tag{3.16}$$

In this model, the vector $\mathbf{w}_{e_{k-1}}$ is an element of a zero-mean, unit-variance, discrete-time, white-process-noise sequence, τ_e is the Markov correlation time constant, σ_e is the steady-state standard deviation for each element in **e**, and I is a $J \times J$ identity matrix. The process noise covariance for the ephemeris errors $Q_{e_{k-1}}$ is equal to the $J \times J$ identity matrix because the intensity scaling is already included in the dynamics mapping.

The ionospheric TEC values are also modeled as evolving according to a first-order Markov process in order to capture their expected time correlations. The model also captures the expected spacial correlation between receivers operating relatively close to one another by dividing the TEC values experienced by a pair of receivers into two parts, an average part, also called the common-mode part, and a differential part.

$$TEC_A^j = TEC_{avg}^j + TEC_{AB}^j \tag{3.17}$$

$$TEC_B^j = TEC_{avg}^j - TEC_{AB}^j \tag{3.18}$$

where TEC_A^j is the TEC along the LOS from receiver A to GPS satellite j, and where TEC_B^j applies for receiver B and GPS satellite j. Here, TEC_{avg}^j is the average TEC between the two receivers and TEC_{AB}^j is the differential TEC. These definitions are illustrated in Fig. 3.2.

The average and differential TEC terms for the J commonly tracked GPS

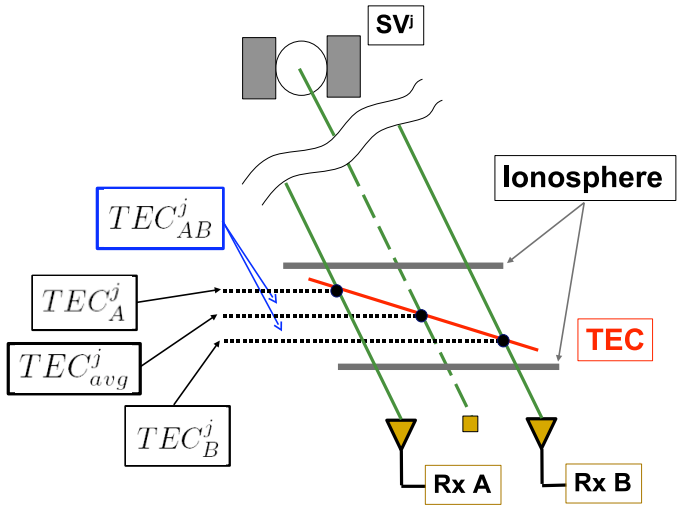

Figure 3.2: Ionosphere model that separates the TEC into average values and differential values. The diagonal line represents the electron density profile across the portion of the ionosphere that is under consideration.

satellites are assembled into the following vector of unknowns:

$$\mathbf{T}_k = \begin{bmatrix} \mathbf{T}_{avg_k} \\ \mathbf{T}_{AB_k} \end{bmatrix} \tag{3.19}$$

where

$$\mathbf{T}_{avg_k} = [TEC^1_{avg_k}\ TEC^2_{avg_k}\ \ldots\ TEC^J_{avg_k}]^T \tag{3.20}$$

$$\mathbf{T}_{AB_k} = [TEC^1_{AB_k}\ TEC^2_{AB_k}\ \ldots\ TEC^J_{AB_k}]^T \tag{3.21}$$

These state definitions allow the noise intensities for the average and differential TEC to be tuned separately in the following TEC dynamics mapping:

$$\mathbf{T}_k = \alpha_{T_{k-1}}\mathbf{T}_{k-1} + \underbrace{\begin{bmatrix} \beta_{T_{avg_{k-1}}} & 0 \\ 0 & \beta_{T_{AB_{k-1}}} \end{bmatrix}}_{\Gamma_{T_{k-1}}} \mathbf{w}_{T_{k-1}} \tag{3.22}$$

where the vector $\mathbf{w}_{T_{k-1}}$ is an element of a zero-mean, unit-variance, white-process-noise sequence. The state mapping and average TEC noise intensity scaling are defined as

$$\alpha_{T_{k-1}} = \exp[-\Delta t_{k-1}/\tau_T] \tag{3.23}$$

$$\beta_{T_{avg_{k-1}}} = \left(\sigma_{T_{avg}} \sqrt{1 - \exp[-2\Delta t_{k-1}/\tau_T]}\right) I \tag{3.24}$$

where τ_T is the correlation time constant, $\sigma_{T_{avg}}$ is the steady-state standard deviation for the average TEC values, and I is a $J \times J$ identity matrix.

Before discussing the differential TEC noise intensity scaling $\beta_{T_{AB_{k-1}}}$, an important distinction between low-altitude and high-altitude CDGPS must be made. At low altitudes, all GPS receivers operate below or inside the ionosphere, and the received signals all pass through a significant portion of the ionosphere. Under these conditions, the effects of TEC can only be neglected when the baseline distances between the receivers are sufficiently small that the differential TEC, the part that influences the relative navigation solution, approaches zero. In GEO and above, many of the received signals never pass through any significant portion of the ionosphere. In these cases, the effects of the differential ionosphere will be very small even over large baseline distances. When the signal's ray path does enter a dense portion of the ionosphere, which is the case when a GPS satellite is rising or setting, large TEC gradients can produce large differential TEC even for two receivers that are relatively close to one another. The differential TEC in these cases must be accurately estimated and removed in order to ensure accurate relative spacecraft state estimates.

The TEC first-order Markov process model incorporates this understanding of the physical situation into the differential TEC noise intensity scaling by considering the receiver-Earth-transmitter geometry. The differential TEC noise

intensity scaling is

$$\beta_{T_{AB_{k-1}}} = \left(\sigma_{T_{AB}} \sqrt{1 - \exp[-2\Delta t_{k-1}/\tau_T]}\right) \begin{bmatrix} \frac{h_{iono}}{d_{los}^1} & & & \\ & \frac{h_{iono}}{d_{los}^2} & & \\ & & \ddots & \\ & & & \frac{h_{iono}}{d_{los}^J} \end{bmatrix} \tag{3.25}$$

The term h_{iono} is the height of the densest portion of the ionosphere and the terms d_{los}^j for $j = 1, 2, \ldots, J$ are the minimum heights of the line-of-sight vectors above the Earth's surface, as defined in Fig. 3.3. As the j^{th} GPS signal's ray path moves close to Earth, d_{los}^j approaches h_{iono}, and the ratio of these values approaches 1. This scenario causes the worst-case noise intensity to be used in the model. Thus, the expected range of values for the differential TEC for channel j will be relatively large, and the filter will be aware of the need to estimate TEC_{AB}^j. On the other hand, as the ray path moves away from Earth, the ratio of h_{iono} to d_{los}^j becomes small, tightening the range of possible values for the differential TEC by reducing the noise intensity. In this situation, the differential TEC should be near zero, and the model informs the filter that an a priori estimate of $TEC_{AB}^j = 0$ is reasonable. Thus, by considering the receiver-Earth-transmitter relative geometry, the filter is aware of the situations in which the differential ionosphere is significant and those in which it is negligible. The process noise covariances for the average TEC, $Q_{T_{avg}}$, and for the differential TEC, $Q_{T_{AB}}$, are both equal to the $J \times J$ identity matrix since the intensity scalings are already included in the TEC dynamics mapping.

Note: it might be reasonable to use a similar altitude-dependent process noise influence for the average TEC values. This level of complexity was not thought worthwhile because of its negligible impact of the relative navigation accuracy.

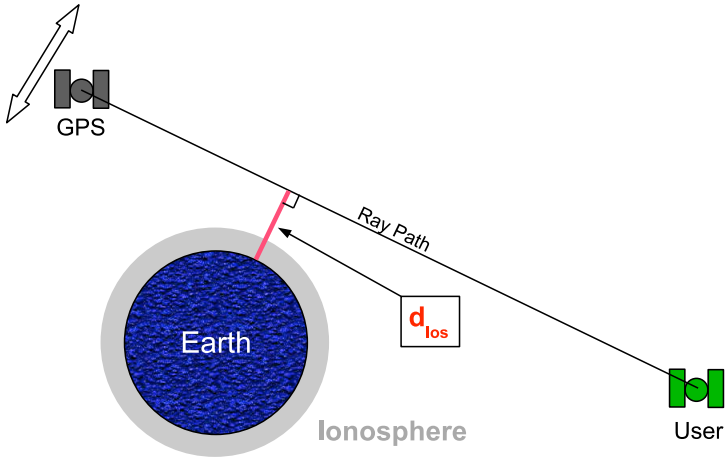

Figure 3.3: Height of ray path above the Earth.

The carrier-phase measurement ambiguities are modeled as being real-valued and exactly constant. They are defined in vector form as follows:

$$\mathbf{a} = \begin{bmatrix} \mathbf{a}_{1A} \\ \mathbf{a}_{1B} \\ \mathbf{a}_{2A} \\ \mathbf{a}_{2B} \end{bmatrix} \tag{3.26}$$

where

$$\mathbf{a}_{1A} = \begin{bmatrix} a_{1A}^{1} \\ a_{1A}^{2} \\ \vdots \\ a_{1A}^{J} \end{bmatrix}; \qquad \mathbf{a}_{1B} = \begin{bmatrix} a_{1B}^{1} \\ a_{1B}^{2} \\ \vdots \\ a_{1B}^{J} \end{bmatrix}; \qquad \mathbf{a}_{2A} = \begin{bmatrix} a_{2A}^{1} \\ a_{2A}^{2} \\ \vdots \\ a_{2A}^{J} \end{bmatrix}; \qquad \mathbf{a}_{2B} = \begin{bmatrix} a_{2B}^{1} \\ a_{2B}^{2} \\ \vdots \\ a_{2B}^{J} \end{bmatrix} \tag{3.27}$$

In these definitions, the superscripts indicate the GPS satellite, the subscript numbers, 1 or 2, indicate the GPS frequency, L1 or L2, and the subscript letters,

A or B, indicate the receiver. Although the ambiguities are real-valued, later developments will exploit the expected integer nature of differences between various of these values.

3.4 Measurement Equations

The measurement equations for the L1-L2 relative navigation filter are presented in this section. The filter uses pseudorange and carrier phase measurements from the L1 and L2 GPS signals. The related L1 pointwise estimator and the related L1 filter only use measurements from the L1 GPS signal and do not estimate TEC.

The standard pseudorange equations are used. These are presented for receiver A at time t_k, although the same equations apply for receiver B. The k subscripts are suppressed for readability.

$$P_{1A}^j = \rho_A^j + c(\delta t_A - \delta t^j) + TEC_A^j + e^j + n_{1A_P}^j \tag{3.28}$$

$$P_{2A}^j = \rho_A^j + c(\delta t_A - \delta t^j) + \frac{f_{L1}^2}{f_{L2}^2} TEC_A^j + e^j + n_{2A_P}^j \tag{3.29}$$

Here, f_{L1} is the nominal L1 frequency, and f_{L2} is the nominal L2 frequency, P_{1A}^j and P_{2A}^j are the L1 and L2 measured pseudoranges from satellite j, ρ_A^j is the geometric range from satellite j, c is the speed of light in a vacuum, δt_A is the receiver clock offset, δt^j is the satellite clock correction, TEC_A^j is the range equivalent TEC at the L1 frequency, and $n_{1A_P}^j$ and $n_{2A_P}^j$ are the errors due to thermal noise and multipath in the L1 and L2 measurements. The carrier-phase measurement equations are based on the model developed in Ref. 23, with the exception that the dependence on the GPS satellite clock correction coefficient a_{f1}

is removed without sacrificing accuracy.

$$\lambda_{L1}\phi_{1A}^{j} = \rho_{A}^{j} + c(\delta t_A - \delta t^j) - TEC_A^j + e^j + \lambda_{L1}a_{1A}^{j} + n_{1A_\phi}^{j} \tag{3.30}$$

$$\lambda_{L2}\phi_{2A}^{j} = \rho_{A}^{j} + c(\delta t_A - \delta t^j) - \frac{f_{L1}^2}{f_{L2}^2}TEC_A^j + e^j + \lambda_{L2}a_{2A}^{j} + n_{2A_\phi}^{j} \tag{3.31}$$

where λ_{L1} is the nominal L1 carrier wavelength, λ_{L2} is the nominal L2 carrier wavelength, a_{1A}^{j} and a_{2A}^{j} are the real-valued L1 and L2 carrier-phase measurement ambiguities as defined in Eq. (3.27), and $n_{1A_\phi}^{j}$ and $n_{2A_\phi}^{j}$ are the errors due to thermal noise and multipath in the L1 and L2 measurements.

It is important to note that the ambiguities in these equations are not integers. Traditional data processing for CDGPS includes the explicit double-differencing of the carrier-phase measurements in order to achieve integers. Double-differenced processing results in correlated measurement errors, makes carrier-phase cycle slip detection and recovery difficult, and discards the knowledge that the real-valued, un-differenced ambiguities are constant. The algorithms in this study attempt to avoid these problems while still taking advantage of the integer nature of the double-differenced ambiguities. They do so in two steps. First, the estimators keep track of the un-differenced ambiguities, and they make sure that the error sources which could make these values appear to be time varying are estimated and removed. Such time-varying error sources include TEC and GPS ephemeris errors. Second, the algorithms exploit the integer nature of the double-differenced ambiguties by transforming the un-differenced ambiguities into double-differenced integer ambiguities during the measurement update step. These integers are estimated using LAMBDA/ILLS techniques and the estimates are used in the navigation solution. A inverse transformation recovers the un-differenced ambiguities without loss of information. These transformations, which were first developed in Ref. 24, are pre-

sented in the Solution Algorithm section.

The traditional CDGPS data processing that explicitly differences the carrier-phase measurements also has the effect of restricting the ways in which the measurements can be used. It has been shown in Ref. 24 that dual-frequency measurements cannot be used simultaneously to do wide laning in order to aid in ambiguity resolution *and* to estimate and remove the effects of the ionosphere. In the traditional approach, the filter designer must choose between alternative differencing schemes in order to achieve these distinct goals and may decide to do wide laning and ionosphere removal in sequence. The approach used in this paper allows the filter to decide how best to use the dual-frequency measurements on a channel-by-channel basis and on-the-fly. This functionality is enabled by giving the filter un-differenced measurements and statistical models of the expected sources of measurement error. With this information, the filter can decide for itself how to weight the alternative types of implicit measurement differencing on a channel-by-channel basis. This "soft" differencing occurs during the square-root information data processing and will be discussed in the next section.

In the following measurement equations, the LOS ephemeris errors to a given GPS satellite are treated as being the same for each receiver at time t_k. An assumption underlies this treatment. It is assumed that the LOS vectors from a particular GPS satellite to the two receivers are similar enough that the projections of the ephemeris errors onto the those vectors are not significantly different. This assumption is reasonable for the types of formations under consideration in this study, but should be reconsidered if the baseline distances become very large, e.g. several hundreds of kilometers.

The measurement equations are linearized about the pseudorange solution at each receiver independently, a method that has proven robust and that encourages fast carrier-phase ambiguity resolution.[15,24] The linearized measurement equations for receiver A are

$$P_{1A}^j - \rho_{A_0}^j + c\delta t^j - (\hat{\rho}_A^j)^T \mathbf{r}_{A_0} = \\ -(\hat{\rho}_A^j)^T \mathbf{r}_A + c\delta t_A + e^j + (TEC_{avg}^j + TEC_{AB}^j) + n_{1A_P}^j \quad (3.32)$$

$$P_{2A}^j - \rho_{A_0}^j + c\delta t^j - (\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_{A_0} = \\ -(\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_A + c\delta t_A + e^j + \frac{f_{L1}^2}{f_{L2}^2}(TEC_{avg}^j + TEC_{AB}^j) + n_{2A_P}^j \quad (3.33)$$

$$\lambda_{L1}\phi_{1A}^j - \rho_{A_0}^j + c\delta t^j - (\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_{A_0} = \\ -(\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_A + c\delta t_A + e^j - (TEC_{avg}^j + TEC_{AB}^j) + \lambda_{L1} a_A^j + n_{1A_\phi}^j \quad (3.34)$$

$$\lambda_{L2}\phi_{2A}^j - \rho_{A_0}^j + c\delta t^j - (\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_{A_0} = \\ -(\hat{\boldsymbol{\rho}}_A^j)^T \mathbf{r}_A + c\delta t_A + e^j - \frac{f_{L1}^2}{f_{L2}^2}(TEC_{avg}^j + TEC_{AB}^j) + \lambda_{L2} a_A^j + n_{2A_\phi}^j \quad (3.35)$$

where $\rho_{A_0}^j$ is the pseudorange solution's range, $\hat{\boldsymbol{\rho}}_A^j$ is the pseudorange solution line-of-sight unit vector defined as pointing from receiver A to GPS satellite j, $\mathbf{r}_{A_0}$ is the pseudorange solution's position estimate defined as $\mathbf{r}_{A_0} = [x_{A_0}(t_k) \;\; y_{A_0}(t_k) \;\; z_{A_0}(t_k)]^T$, and $\mathbf{r}_A$ is the similarly defined true position of the spacecraft. Note: the similar equations apply for receiver B with the exception that the TEC values are $TEC_{avg}^j - TEC_{AB}^j$, consistent with the definition given in Eq. (3.17). The linearization assumes the availability of 4 or more GPS signals for the purposes of calculating the pseudorange solution. It differs from the standard extended Kalman filter technique of linearizing about the a priori state estimate. It is possible, however, that the standard linearization method also would work well.

3.5 Kalman Filter for Mixed Real/Integer States

The relative navigation solution is computed using a square-root information implementation of an extended Kalman filter.[2] This type of implementation is numerically stable and appropriate for use with linear integer estimation techniques that increase the convergence rates to the correct double-differenced carrier-phase ambiguities.

The information equations for the square-root information filter require the inverse square-root of the process noise covariance matrix to be computed. This covariance matrix is a large block matrix that has along its diagonal the individual covariance matrices for each component of the process noise. All of the individual covariance matrices have been defined in the dynamics modeling section. The square-root of the large process noise covariance matrix Q_{k-1} is computed by inverting the transpose of its Cholesky factorization. Thus,

$$R_{ww_{k-1}}^{-1} R_{ww_{k-1}}^{-T} = Q_{k-1} \tag{3.36}$$

where $R_{ww_{k-1}}$ is the desired process noise square-root information matrix.

The dynamics propagation begins by assembling the state and process noise information equations. These equations contain a priori square-root information matrices (SRIMs) and the associated square-root information vectors for each block that will be propagated. These include blocks for the spacecraft position and velocity states and the clock states (combined into one vector $\mathbf{x} = [\mathbf{x}_{sc}^T \quad \mathbf{x}_{clk}^T]^T$), the TEC states, the ephemeris error states, and the real-valued ambiguity states. Using standard SRIF techniques,[2] the information equations

take the following form:

$$\begin{bmatrix} 0 \\ \mathbf{z}_{x_{k-1}} \\ \mathbf{z}_{T_{k-1}} \\ \mathbf{z}_{e_{k-1}} \\ \mathbf{z}_{a_{k-1}} \end{bmatrix} = \begin{bmatrix} R_{ww} & 0 & 0 & 0 & 0 \\ 0 & R_{xx_{k-1}} & R_{xT_{k-1}} & R_{xe_{k-1}} & R_{xa_{k-1}} \\ 0 & 0 & R_{TT_{k-1}} & R_{Te_{k-1}} & R_{Ta_{k-1}} \\ 0 & 0 & 0 & R_{ee_{k-1}} & R_{ea_{k-1}} \\ 0 & 0 & 0 & 0 & R_{aa_{k-1}} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{k-1} \\ \mathbf{x}_{k-1} \\ \mathbf{T}_{k-1} \\ \mathbf{e}_{k-1} \\ \mathbf{a} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{w_{k-1}} \\ \mathbf{v}_{x_{k-1}} \\ \mathbf{v}_{T_{k-1}} \\ \mathbf{v}_{e_{k-1}} \\ \mathbf{v}_{a_{k-1}} \end{bmatrix} \tag{3.37}$$

where the vectors $\mathbf{v}_{w_{k-1}}$, $\mathbf{v}_{x_{k-1}}$, $\mathbf{v}_{T_{k-1}}$, $\mathbf{v}_{e_{k-1}}$, and $\mathbf{v}_{a_{k-1}}$ are all elements of zero-mean, unit-variance, white-noise sequences that are uncorrelated with each other. Since the carrier-phase ambiguities are modeled as being exactly constant from time step to time step, it is not required to include them in the propagation step explicitly. Nevertheless, they are included here for completeness.

The dynamics models in Eqs. (3.6), (3.11), (3.14), and (3.22) are substituted into Eq. (3.37) in order to eliminate the state variables at time t_{k-1} in favor of the state variables at time t_k. The process noise vector $\mathbf{w}_{k-1}$, however, remains in the equation. The resulting equation is subjected to a standard SRIF factorization to produce the following a priori square-root information equation:

$$\begin{bmatrix} \bar{\mathbf{z}}_{w_{k-1}} \\ \bar{\mathbf{z}}_{x_k} \\ \bar{\mathbf{z}}_{T_k} \\ \bar{\mathbf{z}}_{e_k} \\ \bar{\mathbf{z}}_{a_k} \end{bmatrix} = \begin{bmatrix} \bar{R}_{ww_k} & \bar{R}_{wx_k} & \bar{R}_{wT_k} & \bar{R}_{we_k} & \bar{R}_{wa_k} \\ 0 & \bar{R}_{xx_k} & \bar{R}_{xT_k} & \bar{R}_{xe_k} & \bar{R}_{xa_k} \\ 0 & 0 & \bar{R}_{TT_k} & \bar{R}_{Te_k} & \bar{R}_{Ta_k} \\ 0 & 0 & 0 & \bar{R}_{ee_k} & \bar{R}_{ea_k} \\ 0 & 0 & 0 & 0 & \bar{R}_{aa_k} \end{bmatrix} \begin{bmatrix} \mathbf{w}_{k-1} \\ \mathbf{x}_k \\ \mathbf{T}_k \\ \mathbf{e}_k \\ \mathbf{a} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{v}}_{w_{k-1}} \\ \bar{\mathbf{v}}_{p_k} \end{bmatrix} \tag{3.38}$$

where $\bar{\mathbf{v}}_{p_k}$ is an element from a zero-mean, unit-variance, white-noise sequence. At this point, the a priori states could be calculated from Eq. (3.38) by setting $\bar{\mathbf{v}}_{p_k}$ equal to zero and solving by back substitution. This calculation, however, is not required since the measurement linearizations are carried out about the pseudorange solutions rather than the a priori state estimates.

The measurement update is performed next. For square-root information processing, the measurement noise must be characterized in order to precondition the measurement equations so that their Gaussian measurement noise vectors are zero-mean and unit-variance. The measurement error covariance matrices and their square-root inverse factorizations for the pseudorange and carrier-phase measurement equations are

$$P_P = \sigma_P^2 I = R_{vv_P}^{-1} R_{vv_P}^{-T} \tag{3.39}$$

$$P_\phi = \sigma_\phi^2 I = R_{vv_\phi}^{-1} R_{vv_\phi}^{-T} \tag{3.40}$$

where σ_P is the pseudorange measurement error standard deviation and σ_ϕ is the carrier-phase measurement error standard deviation.

The linearized measurements in Eqs. (3.32) through (3.35) are assembled into block form by repeating them for each available GPS satellite and for both receivers. The resulting blocks are each preconditioned by left multiplying them by the inverse square-root of the appropriate measurement noise covariance matrix from Eq. (3.39) or Eq. (3.40). The properly preconditioned measurement equations are written in the following form:

$$\mathbf{y}_k = H_{x_k}\mathbf{x}_k + H_{T_k}\mathbf{T}_k + H_{e_k}\mathbf{e}_k + H_{a_k}\mathbf{a} + \mathbf{n}_k \tag{3.41}$$

This equation is combined with the a proiri information (omitting the process noise blocks) from the propagation step to give the following block form:

$$\begin{bmatrix} \mathbf{y}_k \\ \bar{\mathbf{z}}_{x_k} \\ \bar{\mathbf{z}}_{T_k} \\ \bar{\mathbf{z}}_{e_k} \\ \bar{\mathbf{z}}_{a_k} \end{bmatrix} = \begin{bmatrix} H_{x_k} & H_{T_k} & H_{e_k} & H_{a_k} \\ \bar{R}_{xx_k} & \bar{R}_{xT_k} & \bar{R}_{xe_k} & \bar{R}_{xa_k} \\ 0 & \bar{R}_{TT_k} & \bar{R}_{Te_k} & \bar{R}_{Ta_k} \\ 0 & 0 & \bar{R}_{ee_k} & \bar{R}_{ea_k} \\ 0 & 0 & 0 & \bar{R}_{aa_k} \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{T}_k \\ \mathbf{e}_k \\ \mathbf{a} \end{bmatrix} + \underbrace{\begin{bmatrix} \mathbf{n}_k \\ \bar{\mathbf{v}}_{p_k} \end{bmatrix}}_{\bar{\mathbf{v}}_{m_k}} \tag{3.42}$$

where $\bar{\mathbf{v}}_{m_k}$ is a zero-mean, unit-variance noise vector implicitly defined in this equation.

Standard SRIF factorization techniques are used to transform the Eq. (3.42) into the following block upper triangular form:

$$\begin{bmatrix} \mathbf{z}_{x_k} \\ \mathbf{z}_{T_k} \\ \mathbf{z}_{e_k} \\ \mathbf{z}_{a_k} \\ \mathbf{z}_{r_k} \end{bmatrix} = \begin{bmatrix} R_{xx_k} & R_{xT_k} & R_{xe_k} & R_{xa_k} \\ 0 & R_{TT_k} & R_{Te_k} & R_{Ta_k} \\ 0 & 0 & R_{ee_k} & R_{ea_k} \\ 0 & 0 & 0 & R_{aa_k} \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{T}_k \\ \mathbf{e}_k \\ \mathbf{a} \end{bmatrix} + \mathbf{v}_{m_k} \tag{3.43}$$

where $\mathbf{z}_{r_k}$ is the square-root information measurement residual vector.

The un-differenced, real-valued ambiguity block is isolated in Eq. (3.43), and a transformation is developed that separates the ambiguities on each frequency into irreducibly real elements and double-differenced integer elements. This transformation allows the integer parts to be resolved using powerful integer least-squares techniques. This transformation is equivalent to the one developed in Ref. 24. The double-differenced integer ambiguities are defined as follows:

$$\mathbf{N} = \begin{bmatrix} (a^2_{1B} - a^2_{1A}) - (a^1_{1B} - a^1_{1A}) \\ (a^3_{1B} - a^3_{1A}) - (a^1_{1B} - a^1_{1A}) \\ \vdots \\ (a^J_{1B} - a^J_{1A}) - (a^1_{1B} - a^1_{1A}) \\ (a^2_{2B} - a^2_{2A}) - (a^1_{2B} - a^1_{2A}) \\ (a^3_{2B} - a^3_{2A}) - (a^1_{2B} - a^1_{2A}) \\ \vdots \\ (a^J_{2B} - a^J_{2A}) - (a^1_{2B} - a^1_{2A}) \end{bmatrix} = G\mathbf{a} \tag{3.44}$$

where G_k is a $2(J-1) \times 4J$ matrix of ones, negatives ones, and zeros arranged

to produce $\mathbf{N}$, which is a $2(J-1)\times 1$ vector of double-differenced integer ambiguities. The transformation matrix is factorized into the product of a square, orthonormal matrix and an upper-triangular matrix, a factorization commonly called QR factorization,[8] resulting in the following system:

$$Q_1 \begin{bmatrix} R_1^T \\ 0 \end{bmatrix} = G^T \tag{3.45}$$

or transposing

$$[R_1 \quad 0]\, Q_1^T = G \tag{3.46}$$

Here, Q_1 is the orthonormal matrix and R_1^T is a square, non-singular, upper triangular matrix. This factorization may be used to define the following underdetermined set of equations for the real-valued ambiguities give the double-differenced ambiguities:

$$\mathbf{N} = [R_1\ 0]Q_1^T\mathbf{a} \tag{3.47}$$

The underdetermined subspace of $\mathbf{a}$ can be defined as a vector of irreducibly real ambiguities $\mathbf{a}_{ir}$ that has dimension $2(J+1)$. The relationship between this new vector, $\mathbf{a}_{ir}$, the original vector, $\mathbf{a}$, and the integer ambiguity vector, $\mathbf{N}$, is

$$\begin{bmatrix} \mathbf{a}_{ir} \\ \mathbf{N} \end{bmatrix} = \begin{bmatrix} 0 & I \\ R_1^T & 0 \end{bmatrix} Q_1^T\mathbf{a} \tag{3.48}$$

Solving for $\mathbf{a}$ yields

$$\mathbf{a} = Q_1 \begin{bmatrix} 0 & R_1^{-T} \\ I & 0 \end{bmatrix} \begin{bmatrix} \mathbf{a}_{ir} \\ \mathbf{N} \end{bmatrix} = E \begin{bmatrix} \mathbf{a}_{ir} \\ \mathbf{N} \end{bmatrix} \tag{3.49}$$

where the $2(J+1)\times 2(J+1)$ matrix E is implicitly defined by this equation.

In order to apply LAMBDA/ILLS methods to determine the optimal estimate of $\mathbf{N}$, it is necessary to develop an information equation that isolates $\mathbf{N}$

from the real-valued estimated states. This is don by using Eq. (3.49) to eliminate **a** from Eq. (3.43) in favor of the unknowns $\mathbf{a}_{ir}$ and **N**. Afterwards, the fourth row of the information equation is transformed using QR factorization to isolate an **N** information equation. The QR factorization and transformation to do this take the form

$$Q_{2_k}\begin{bmatrix} R_{irir_k} & R_{irN_k} \\ 0 & R_{NN_k} \end{bmatrix} = R_{aa_k}E \tag{3.50}$$

$$\begin{bmatrix} \mathbf{z}_{ir_k} \\ \mathbf{z}_{N_k} \end{bmatrix} = Q_{2_k}^T \bar{\mathbf{z}}_{a_k} \tag{3.51}$$

where Q_{2_k} is an orthonormal matrix, R_{irir_k} and R_{NN_k} are square, non-singular, upper triangular matrices, and R_{irN_k} is a dense matrix of appropriate dimensions. Some additional transformations that will aid in the computation of the state estimates and estimation error covariance matrices are performed:

$$\begin{bmatrix} R_{xir_k} & R_{xN_k} \\ R_{Tir_k} & R_{TN_k} \\ R_{eir_k} & R_{eN_k} \end{bmatrix} = \begin{bmatrix} R_{xa_k} \\ R_{Ta_k} \\ R_{ea_k} \end{bmatrix} E \tag{3.52}$$

At this point, the following a posteriori square-root information equation applies:

$$\begin{bmatrix} \mathbf{z}_{x_k} \\ \mathbf{z}_{T_k} \\ \mathbf{z}_{e_k} \\ \mathbf{z}_{ir_k} \\ \mathbf{z}_{N_k} \\ \mathbf{z}_{r_k} \end{bmatrix} = \begin{bmatrix} R_{xx_k} & R_{xT_k} & R_{xe_k} & R_{xir_k} & R_{xN_k} \\ 0 & R_{TT_k} & R_{Te_k} & R_{Tir_k} & R_{TN_k} \\ 0 & 0 & R_{ee_k} & R_{eir_k} & R_{eN_k} \\ 0 & 0 & 0 & R_{irir_k} & R_{irN_k} \\ 0 & 0 & 0 & 0 & R_{NN_k} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{x}_k \\ \mathbf{T}_k \\ \mathbf{e}_k \\ \mathbf{a}_{ir} \\ \mathbf{N} \end{bmatrix} + \mathbf{v}_{m_k} \tag{3.53}$$

where $\mathbf{v}_{m_k}$ is the appropriately transformed zero-mean, unit-variance noise vector. Notice how the integer ambiguity states, **N**, in the lower, right-hand block of

Eq. (3.53) are isolated from the real-valued states. This decoupling of the integer problem allows specialized techniques to be used to solve for $\mathbf{N}$.

The ambiguity SRIM R_{NN_k} and the ambiguity information vector $\mathbf{z}_{N_k}$ are passed to a decorrelation and integer linear least-squares solution algorithm which finds the $\mathbf{N}$ that minimizes

$$J(\mathbf{N}) = \frac{1}{2}(R_{NN_k}\mathbf{N} - \mathbf{z}_{N_k})^T(R_{NN_k}\mathbf{N} - \mathbf{z}_{N_k}) \tag{3.54}$$

under the constraint that

$$\mathbf{N} \in Z^{2(J-1)} \tag{3.55}$$

where $Z^{2(J-1)}$ indicates the $2(J-1)$-dimensional integer vector space. The decorrelation part of this process is merely a preconditioning step that further transforms R_{NN_k} and $\mathbf{z}_{N_k}$ in order to expedite the integer least-squares optimization. Algorithms that perform this decorrelation and optimization are commonly referred to as LAMBDA/ILLS methods. For more information on the LAMBDA/ILLS algorithms used in this paper, please see Ref. 22.

Once the double-differenced integer ambiguities have been estimated,[†] the measurement update step is completed by calculating the remaining a posteriori state estimates via back substitution in Eq. (3.53):

$$\hat{\mathbf{a}}_{ir} = R^{-1}_{irir_k}[\mathbf{z}_{ir_k} - R_{irN_k}\mathbf{N}_{opt}] \tag{3.56}$$

$$\hat{\mathbf{e}}_k = R^{-1}_{ee_k}[\mathbf{z}_{e_k} - R_{eir_k}\hat{\mathbf{a}}_{ir} - R_{eN_k}\mathbf{N}_{opt}] \tag{3.57}$$

$$\hat{\mathbf{T}}_k = R^{-1}_{TT_k}[\mathbf{z}_{T_k} - R_{Te_k}\hat{\mathbf{e}}_{k-1} - R_{Tir_k}\hat{\mathbf{a}}_{ir} - R_{TN_k}\mathbf{N}_{opt}] \tag{3.58}$$

$$\hat{\mathbf{x}}_k = R^{-1}_{xx_k}[\mathbf{z}_{x_k} - R_{xT_k}\hat{\mathbf{T}}_{k-1} - R_{xe_k}\hat{\mathbf{e}}_{k-1} - R_{xir_k}\hat{\mathbf{a}}_{ir} - R_{xN_k}\mathbf{N}_{opt}] \tag{3.59}$$

[†]Note: the Kalman filter should also validate its double-differenced integer ambiguity estimates. Validation is the process of determining whether the probability that $\mathbf{N}_{opt}$ is correct is sufficiently high to assume the value to be exact. For more information on ambiguity validation, please see Ref. 27. At present, the Kalman filter does not perform a validation calculation. Given that it re-estimates the integers at each measurement update, this can be an acceptable approach because it never strictly fixes the values of $\mathbf{N}$.

The vectors $\mathbf{z}_{x_k}$, $\mathbf{z}_{T_k}$, $\mathbf{z}_{e_k}$ and $\mathbf{z}_{a_k}$, and the matrices R_{xx_k}, R_{xT_k}, R_{xe_k}, R_{xa_k}, R_{TT_k}, R_{Te_k}, R_{Ta_k}, R_{ee_k}, R_{ea_k}, and R_{aa_k} from Eq. (3.43) are all passed forward to the filter's next recursion for use in the dynamics propagation step. These square-root information vectors and matrices, however, are only valid if the measurements used in the filter's next recursion are taken from the same set of GPS satellites used in its current recursion. If any signals are added or dropped, then the square-root information vectors and matrices in Eq. (3.43) must be modified.

Consider the case where a new satellite is acquired on the p^{th} channel. Seven new variables must be estimated: the average and differential total electron content along the signals' ray path, TEC^p_{avg} and TEC^p_{AB}; the new GPS satellite's ephemeris error, e^p; and the real-valued, un-differenced ambiguities for each receiver and each GPS frequency, a^p_{1A}, a^p_{2A}, a^p_{1B}, and a^p_{2B}. The state vector from Eq. (3.43) is augmented with these new variables, and the square-root information vectors and matrices from Eq. (3.43) are augmented to include a reasonable a priori square-root information equation for each new variable. For the TEC and ephemeris error variables, the new square-root information equations take the following form:

$$z^p_{T_{avg}} = \left(\frac{1}{\sigma_{T_{avg}}}\right) TEC^p_{avg} + v^p_{T_{avg}} \tag{3.60}$$

$$z^p_{T_{AB}} = \left(\frac{1}{\sigma_{T_{AB}}}\right) TEC^p_{AB} + v^p_{T_{AB}} \tag{3.61}$$

$$z^p_e = \left(\frac{1}{\sigma_e}\right) e^p + v^p_e \tag{3.62}$$

In these equations, the terms on the left-hand side, $z^p_{T_{avg}}$, $z^p_{T_{AB}}$, and z^p_e, are the a priori square-root information elements to be inserted into the existing square-root information vectors. The terms $\sigma_{T_{avg}}$, $\sigma_{T_{AB}}$, and σ_e are the expected standard deviations of the new variables, and the reciprocals of these terms, shown in parentheses in Eqs. (3.60) through (3.62), are the new diagonal elements to be

inserted into the existing SRIMs. The terms $v^p_{T_{avg}}$, $v^p_{T_{AB}}$, and v^p_e are the new elements of the noise vectors.

The goal here is to provide the filter with appropriate a priori statistics for the new variables. This is done by choosing reasonable values for $z^p_{T_{avg}}$, $z^p_{T_{AB}}$, z^p_e, $\sigma_{T_{avg}}$, $\sigma_{T_{AB}}$, and σ_e based on the physics of the navigation problem. Since the TEC values in the high-altitude orbits are usually near zero, and since the ephemeris errors are expected to be zero mean, a reasonable choice for the a priori values of $z^p_{T_{avg}}$, $z^p_{T_{AB}}$, and z^p_e is to set each equal to zero. The choice for the a priori standard deviation values is guided by the Markov models described above. The maximum standard deviations in those models are used for $\sigma_{T_{avg}}$, $\sigma_{T_{AB}}$, and σ_e.

The square-root information equations for the remaining four new variables, the new ambiguities, take a similar form:

$$z^p_{a_{1A}} = \left(\frac{1}{\sigma_a}\right) a^p_{1A} + v^p_{a_{1A}} \tag{3.63}$$

This equation applies for the L1 ambiguity at receiver A, although similar equations apply for the L2 ambiguity at receiver A, and for the L1 and L2 ambiguities at receiver B. Since the new ambiguity in Eq. (3.63) can take on any value, the a priori uncertainty associated with it is infinite, i.e. $\sigma_a = \infty$. The infinite uncertainty indicates that $z^p_{a_{1A}}$ contains no useful information for the filter, and that it may, therefore, be set to an arbitrary value. Notice the assumption in Eqs. (3.60) through (3.63) that the new unknowns are uncorrelated with each other and with the previously estimated unknowns. This assumption makes augmenting the appropriate blocks in Eq. (3.43) a straightforward task.

Next, consider the case where the p^{th} commonly tracked GPS satellite is dropped. The same seven variables discussed in the previous paragraphs and

their associated square-root information equations are now discarded. The dropped variables, however, are not independent of the retained variables. In other words, the square-root information equations associated with the dropped variables also contain information about the retained variables. A procedure must be applied that separates the information that applies to the kept variables from the information that applies to the dropped variables, so that the latter may be discarded without losing useful information. This procedure involves three steps. First, the state vector is permuted in order to place the dropped variables at the top of the vector. This has the effect of reordering the columns of the large block upper triangular SRIM in Eq. (3.43) such that the columns associated with the dropped variables are moved to the first seven columns of the matrix. The modified information equation, with the time index subscripts suppressed for readability, is

$$\begin{bmatrix} \mathbf{z}_x \\ \mathbf{z}_T \\ \mathbf{z}_e \\ \mathbf{z}_a \\ \mathbf{z}_r \end{bmatrix} = \begin{bmatrix} R_{xd} & R_{xx} & \tilde{R}_{xT} & \tilde{R}_{xe} & \tilde{R}_{xa} \\ R_{Td} & 0 & \tilde{R}_{TT} & \tilde{R}_{Te} & \tilde{R}_{Ta} \\ R_{ed} & 0 & 0 & \tilde{R}_{ee} & \tilde{R}_{ea} \\ R_{ad} & 0 & 0 & 0 & \tilde{R}_{aa} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{x} \\ \tilde{\mathbf{T}} \\ \tilde{\mathbf{e}} \\ \tilde{\mathbf{a}} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_d \\ \tilde{\mathbf{v}}_{m_k} \end{bmatrix} \tag{3.64}$$

where

$$\mathbf{d} = \begin{bmatrix} TEC_{avg}^{p} \\ TEC_{AB}^{p} \\ e^{p} \\ a_{1A}^{p} \\ a_{2A}^{p} \\ a_{1B}^{p} \\ a_{2B}^{p} \end{bmatrix} \tag{3.65}$$

and where R_{xd}, R_{Td}, R_{ed}, and R_{ad} are constructed by taking the columns from

Eq. (3.43) that are associated with the dropped variables and grouping them together. The tilde notation as applied to a SRIM indicates that the matrix has been modified by removing one or more of its columns. The tilde notation as applied to a state vector or a noise vector indicates that particular elements of the vector have been removed. The second step in the dropped satellite procedure is to decouple the information associated with the retained variables from the information associated with the dropped variables. This decoupling is accomplished by QR factorizing and transforming Eq. (3.64), resulting in the following block upper triangular system:

$$\begin{bmatrix} \bar{\mathbf{z}}_d \\ \bar{\mathbf{z}}_x \\ \bar{\mathbf{z}}_T \\ \bar{\mathbf{z}}_e \\ \bar{\mathbf{z}}_a \\ \mathbf{z}_r \end{bmatrix} = \begin{bmatrix} \bar{R}_{dd} & \bar{R}_{dx} & \bar{R}_{dT} & \bar{R}_{de} & \bar{R}_{da} \\ 0 & \bar{R}_{xx} & \bar{R}_{xT} & \bar{R}_{xe} & \bar{R}_{xa} \\ 0 & 0 & \bar{R}_{TT} & \bar{R}_{Te} & \bar{R}_{Ta} \\ 0 & 0 & 0 & \bar{R}_{ee} & \bar{R}_{ea} \\ 0 & 0 & 0 & 0 & \bar{R}_{aa} \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d} \\ \mathbf{x} \\ \tilde{\mathbf{T}} \\ \tilde{\mathbf{e}} \\ \tilde{\mathbf{a}} \end{bmatrix} + \begin{bmatrix} \bar{\mathbf{v}}_d \\ \bar{\mathbf{v}}_m \end{bmatrix} \tag{3.66}$$

The final step is to discard the first row of this equation and to extract the proper information vectors and matrices to pass forward to the next recursion. These vectors and matrices are taken from the second through fifth rows of Eq. (3.66). Note: if the double-differenced integer ambiguities for the dropped satellite had been validated, then a more complex satellite dropping precedure could have been developed in order to retain the information that is inherent in the validation. Also, the add/drop procedures described in this section have been presented for adding/dropping a single GPS satellite. The methods, however, may be applied recursively to handle adding/dropping multiple satellites.

3.6 Results

The relative navigation algorithms have been tested using Monte Carlo simulations. Each Monte Carlo run uses a high-fidelity truth-model simulation that replicates the output of GPS receivers' pseudorange and carrier-phase measurements in user defined satellite formation scenarios. The off-line MATLAB simulation algorithm is described in detail in Ref. 23. It incorporates physics-based models of many error sources: the effects of the ionosphere and multipath; antenna phase-center variations and polarization-induced phase windup; receiver clock drift, thermal noise, and line biases; and GPS satellite ephemeris errors. For details about these error models, please consult Ref. 23.

Two template scenarios have been designed to test a filter's navigation performance in high-altitude orbits. The first consists of a two-spacecraft formation orbiting in GEO with a nominal baseline distance of 3 *km*. The simulated receivers are assumed to have the weak-signal acquisition and tracking capability described in Refs. 19 and 21. Using such technology, receivers in GEO are assumed to be able to acquire and track signals with a C/N_0 as low as 18 dB-Hz. This level of sensitivity allows the receivers to track up to 12 GPS signals with GDOP values ranging between 5 and 15.[23] These scenarios are referred to as the GEO scenarios. The second template scenario consists of a two-spacecraft formation near apogee in a 1.2×18 Earth radii orbit with a nominal baseline distance of 10 km. The simulated receivers are assumed to be able to track signals down to $C/N_0 = 12$ dB-Hz. This threshold allows the tracking of as many as 11 satellites at apogee with GDOP values ranging between 110 and 300.[23] These scenarios are referred to as the HEO scenarios.

The performance of the L1-L2 filter, the L1-only filter, and the L1-only pointwise estimator are compared by examining specific Monte Carlo runs that stress the estimators in particular ways and by examining the statistical performance over all of the Monte Carlo runs. Four representative classes of scenarios are considered. The first class includes relatively short duration scenarios (several hours long) in which nothing unusual happens. Scenarios in this class will be referred to as "quiescent" scenarios. The second class includes short duration scenarios in which there is a significant change in the differential ionosphere on one or more of the channels. Such a change occurs when a GPS satellite is rising or setting behind the Earth. Scenarios in this class will be referred to as "ionosphere event" scenarios. The third class includes short scenarios that challenge the estimators' ability to converge and to remain locked onto the correct integer ambiguities. Differential ionosphere, poor GDOP, slow line-of-sight vector dynamics, and noisy measurements all may stress the estimators' convergence and robustness. Scenarios in this class will be referred to as "convergence/robustness" scenarios. The fourth class includes long duration scenarios lasting approximately one orbit, or 24 hours in GEO and 38 hours in HEO. Scenarios in this class will be referred to as "full orbit" scenarios.

The tuning parameters for the dynamics models are presented in Table 3.1. In the table, $\beta_{w_{AB}}$ is the common-mode process noise variance and $\beta_{\delta w_{AB}}$ is the differential-mode variance for the spacecraft orbital dynamics model. The terms h_0 and h_{-2} are typical Allen variances for a temperature compensated crystal oscillator. Next, σ_e is the steady-state standard deviation, and τ_e is the correlation time constant for the ephemeris error Markov model. The term $\sigma_{T_{avg}}$ is the common-mode steady-state standard deviation, $\sigma_{T_{AB}}$ is the differential-mode standard deviation, and τ_T is the correlation time constant for the TEC

Markov model. The measurement error standard deviations are assumed to be $\sigma_P = 7\ m$ for the pseudorange and $\sigma_\phi = 0.01\ cycles$ for the carrier phase.

Table 3.1: Tuning parameters in GEO and HEO scenarios

States	Symbols	L1-L2 Filter	L1 Filter	L1 Pointwise
Spacecraft	$\beta_{w_{AB}}\ (m^2/s^3)$	3×10^{-2}	3×10^{-2}	-
	$\beta_{\delta w_{AB}}\ (m^2/s^3)$	5×10^{-7}	5×10^{-7}	-
Receiver Clocks	$h_0\ (m^2/s)$	$(7 \times 10^{-21}) \times c^2$	$(7 \times 10^{-21}) \times c^2$	-
	$h_{-2}\ (m^2/s^3)$	$(2 \times 10^{-20}) \times c^2$	$(2 \times 10^{-20}) \times c^2$	-
Ephemeris Errors	$\sigma_e\ (m)$	1	1	1
	$\tau_e\ (s)$	20000	20000	20000
TEC	$\sigma_{T_{avg}}\ (m)$	5	-	-
	$\sigma_{T_{AB}}\ (m)$	0.05	-	-
	$\tau_T\ (s)$	10	-	-

The performance of the estimators in a quiescent GEO scenario is illustrated in Fig. 3.4, which plots a 3.5 hour time history of the 3-dimensional relative position error components. The error components are expressed in a coordinate frame that is centered at one of the GPS receiver's antenna. Its axes point along the radial direction from the center of the Earth (radial), along the direction of motion (along track), and along the direction normal to the orbit plane (cross track). The plot shows that all three estimators perform well in this particular scenario. They all converge to the correct integer ambiguities in one measurement step, and navigate to sub-decimeter-level accuracy or better. The L1-only pointwise estimator experiences maximum errors of about 5-7 cm, and the L1-

only and L1-L2 filters experience maximum errors of about 2-3 cm. As expected, the filtered solutions are less noisy than the pointwise solution. The comparable performance of the L1-only filter and the L1-L2 filter is also expected given the fact that the differential ionosphere is negligible throughout the scenario. When the quiescent periods of 100 Monte Carlo runs are considered, the L1-only pointwise estimator has a mean error magnitude of 9.2 cm and a maximum error magnitude of 11.0 cm, the L1-only filter has a mean error magnitude of 1.1 cm and a maximum error magnitude of 6.0 cm, and the L1-L2 filter has a mean error magnitude of 1.2 cm and a maximum error magnitude of 6.0 cm.

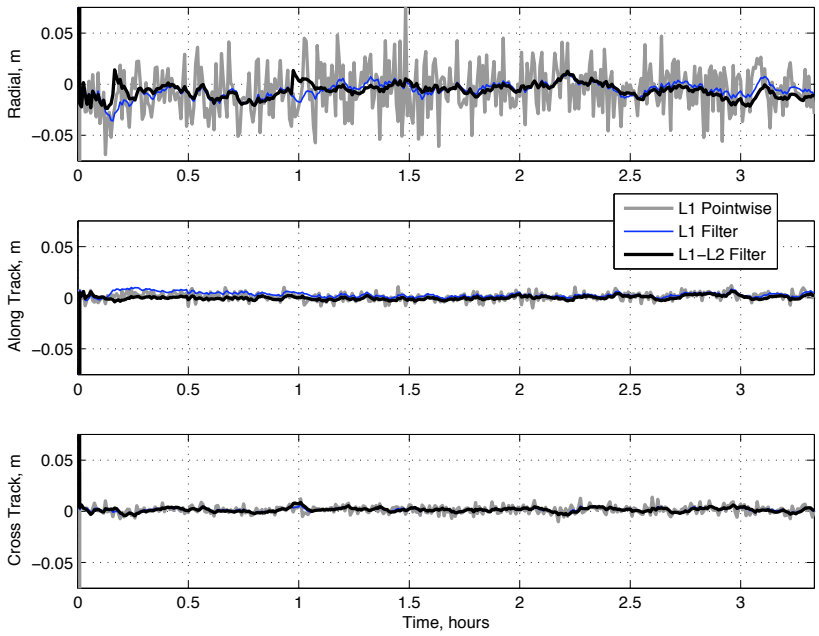

Figure 3.4: Relative position error components in a quiescent GEO scenario.

All three estimators also perform well in the quiescent HEO scenarios. Figure 3.5 shows that, for a particular quiescent scenario, all the estimators converge to the correct integer ambiguities immediately. The L1-only pointwise estimator navigates to within about 1-1.5 m of the true relative position, and

the filters, after some initial transients, navigate to within 20 cm of the truth. The relatively large errors in the filters' solutions over the first 0.5 hours are due to the poor initial observability of the ephemeris error and TEC states. After a sufficient amount of data is processed, the Markov models converge, and the navigation errors settle to small values. Again, the L1-only and L1-L2 filters perform comparably when the effects of differential ionosphere are negligible. When the quiescent periods of 100 Monte Carlo runs are considered, the L1-only pointwise estimator has a mean error magnitude of 43 cm and a maximum error magnitude of 5 m, the L1-only filter has a mean error magnitude of 13 cm and a maximum error magnitude of 54 cm, and the L1-L2 filter has a mean error magnitude of 11 cm and a maximum error magnitude of 54 cm. The quiescent scenario results are summarized in Table 3.2.

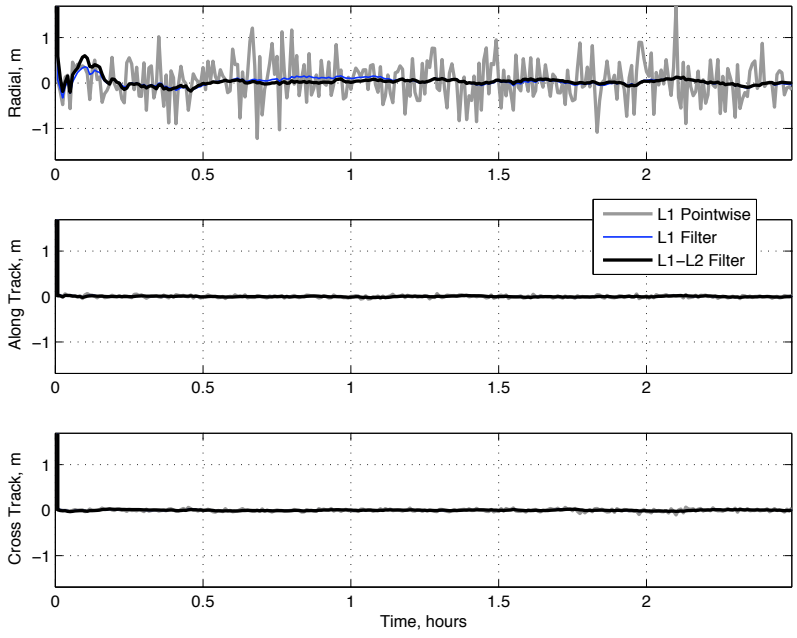

Figure 3.5: Relative position error components in a quiescent HEO scenario.

The performance of the estimators in a particular ionosphere event scenario

Table 3.2: Relative position error magnitudes for the quiescent scenarios

		L1 Pointwise	L1 Filter	L1-L2 Filter
GEO	mean error (cm)	9.2	1.1	1.2
	maximum error (cm)	11.0	6.0	6.0
HEO	mean error (cm)	43	13	11
	maximum error (cm)	500	54	54

is shown in Fig. 3.6 for the GEO formation. The L1-only filter and the L1-only pointwise estimator both show a significant increase in error about halfway through the scenario when one of the tracked GPS satellites sets behind the Earth. Moments before the signal is lost, the ray path dips into a dense portion of the ionosphere, increasing the absolute TEC experienced at each receiver and, more importantly, increasing the differential TEC experienced by the receivers. The latter effect is caused by the large TEC gradients to which the signal is subjected as it drops into the ionosphere. To get a better sense of how the signal interacts with the ionosphere, consider Fig. 3.7 which plots the orthogonal height of the setting satellite's ray path, a quantity that has been defined in Fig. 3.3. The ray path height drops from 7000 km above the Earth down to 350 km, the height at which the simulator drops the signal. That height is also the height of the densest portion of the simulated ionosphere. As the satellite sets, the pointwise estimator experiences a larger peak error, about 1 m, but recovers as soon as the signal is dropped. The L1-only filter has a smaller peak error, about 0.5 m, but takes longer to recover due to the averaging effects of filtering. The important result here, however, is that the dual-frequency filter successfully estimates the differential TEC and removes its effects. Its relative navigation solution shows

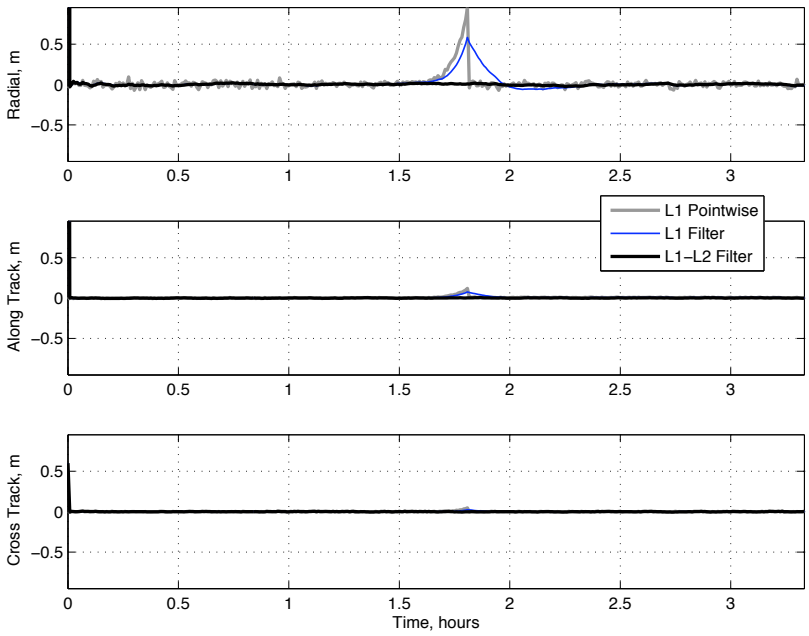

Figure 3.6: Relative position error components in an ionosphere event GEO scenario.

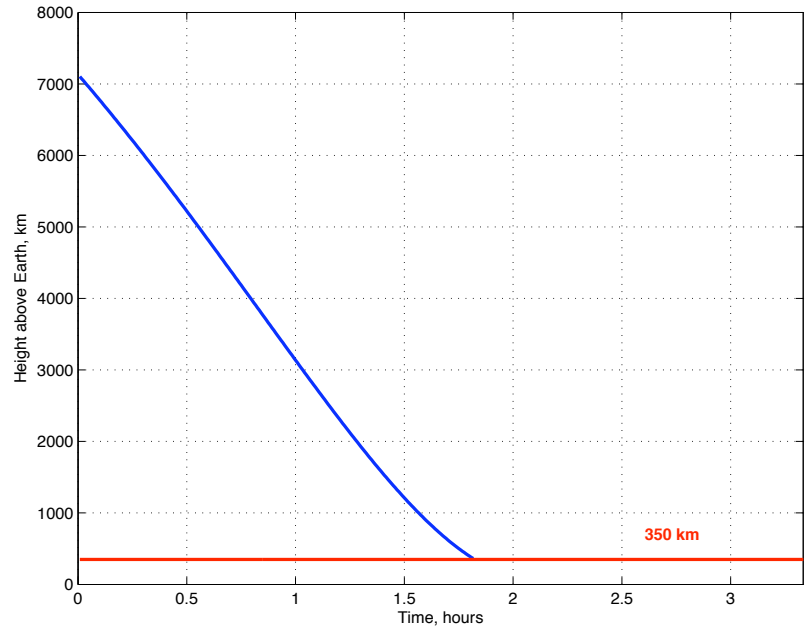

Figure 3.7: Ray path height of a setting GPS satellite's signal.

no detectable degradation during the occultation. Considering the ionosphere events of 100 Monte Carlo runs, the L1-only pointwise estimator has a peak error magnitude of 2.03 m, the L1-only filter has a peak error magnitude of 1.99 m, and the L1-L2 filter has a peak error magnitude of 6 cm. Notice that the L1-L2 filter has exactly the same error magnitude as it had during the quiescent scenarios.

The performance of the navigation algorithms in the HEO scenarios during ionosphere events is similar to the performance in the GEO scenarios, with the exception that the magnitude of the errors is much larger. Considering the ionosphere events of 100 Monte Carlo runs, the L1-only pointwise estimator has a peak error magnitude of 28.70 m, the L1-only filter has a peak error magnitude of 4.41 m, and the L1-L2 filter has a peak error magnitude of 54 cm. Again, the L1-L2 filter's peak error magnitude identical to its quiescent scenario peak error magnitude. One could avoid the differential-ionosphere-induced navigation problems apparent in the L1-only estimators by discarding data from signals that pass too near the Earth. Those signals, however, are the strongest available at any given time and, therefore, are the easiest to acquire and track. Considering the difficulty of signal reception at high altitudes, one should hesitate before discarding any data. The ionosphere event scenario results are summarized in Table 3.3.

The convergence/robustness of the estimators is best illustrated by considering a particular HEO scenario that stresses the ability of the single-frequency estimators to estimate the correct integer ambiguities. Figure 3.8 shows a 60 minute time history of the relative position error in such a scenario. Notice how the L1-only filter and L1-only pointwise estimator both take about 15 min-

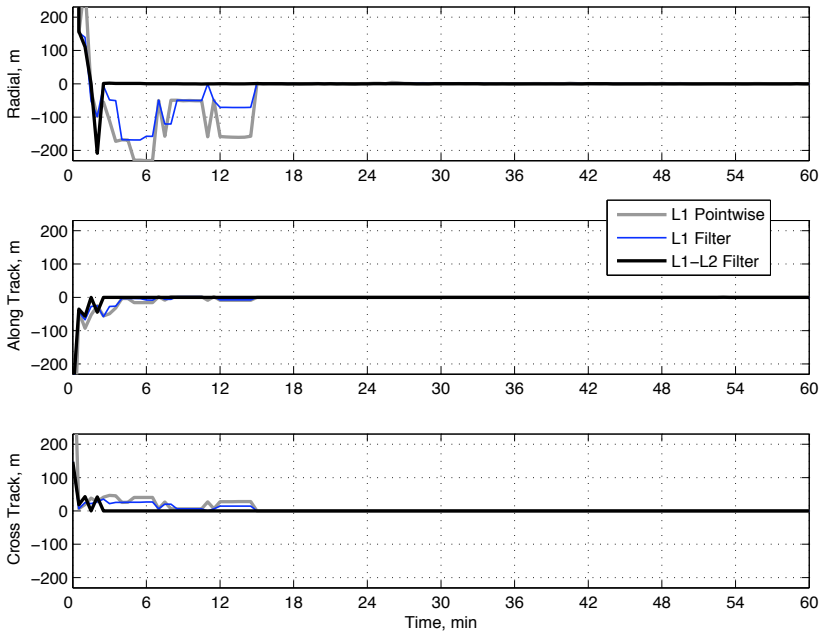

Figure 3.8: Relative position error components for a HEO scenario that challenges the estimators' convergence capabilities.

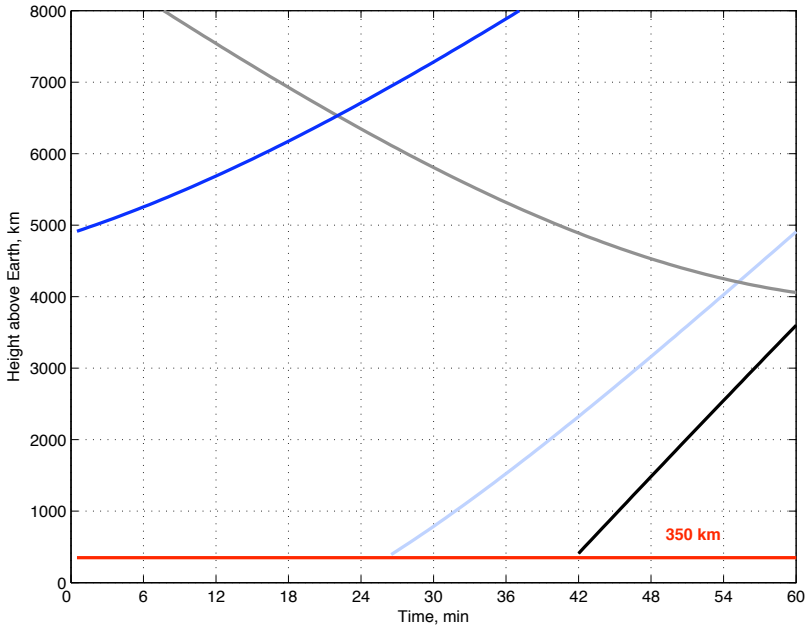

Figure 3.9: Ray path height for several signals during a HEO scenario that challenges the estimators' convergence capabilities.

Table 3.3: Relative position error magnitude for the ionosphere event scenarios

		L1 Pointwise	L1 Filter	L1-L2 Filter
GEO	maximum error (m)	2.03	1.99	0.06
HEO	maximum error (m)	28.7	4.41	0.54

utes to converge to the correct integers, whereas the L1-L2 filter takes less than 3 minutes to converge. This behavior, where one or both of the L1-only estimators converges slower than does the L1-L2 filter, occurs in 44% of the HEO Monte Carlo runs and can be attributed to ionosphere effects in most cases. In this particular scenario, however, the slow convergence cannot be attributed to the ionosphere since all of the ray paths are well above the dense portion of the ionosphere during the initial convergence period, as illustrated by the first 15 minutes of the graph in Fig. 3.9. This result indicates that, in the absence of ionosphere effects, the L1-L2 filter is using the dual-freqency measurements to aid in integer ambiguity resolution by performing implicit wide laning rather than using the measurements to remove the effects of the ionosphere. The choice of how to best use the measurements can be made on a channel-by-channel basis, allowing the filter to use dual-frequency measurements to remove the effects of the ionosphere on some channels, while doing wide laning on other channels. This multi-modal capability may account for the large improvements in covergence/robustness performance exhibited by the L1-L2 filter.

In the GEO convergence/robustness Monte Carlo runs, the L1-only pointwise estimator requires a mean of 1.06 thirty-second measurement steps to converge and requires a maximum of 51 steps, and the L1-only filter requires a

mean of 1.59 steps and a maximum of 51 steps. The L1-L2 filter converges to the correct integer ambiguities in the first measurement step in all Monte Carlo runs. In the HEO runs, the L1-only pointwise estimator converges in mean of 9.4 thirty-second measurement steps and in 2 cases fails to converge in the 2.5 hour scenarios. The L1-only filter converges in a mean of 10.5 steps and also fails to converge in 2 scenarios. The L1-L2 filter converges in a mean of 1.54 steps and a maximum of 11 steps. The convergence results are summarized in Table 3.4.

Table 3.4: The number of 30-second measurement steps required to converge to the correct integer ambiguities

		L1 Pointwise	L1 Filter	L1-L2 Filter
GEO	mean (steps)	1.06	1.59	1
	maximum (step)	51	51	1
HEO	mean (steps)	9.4	10.5	1.54
	maximum (steps)	>300	>300	11

The robustness of the L1-L2 filter is highlighted again in the full orbit scenarios. Figure 3.10 shows one full orbit in a particular GEO scenario. The single-frequency estimators are relatively slow to converge to the correct ambiguities, and they experience large spikes in error periodically due to increases in the differential TEC that occurs when GPS satellites set. The dual-frequency filter converges immediately and experiences no degradation of the solution when satellites set.

The L1-L2 filter's robustness is also apparent in the full orbit HEO scenarios. In these 38 hour simulations, the formation satellites begin and end at an apogee

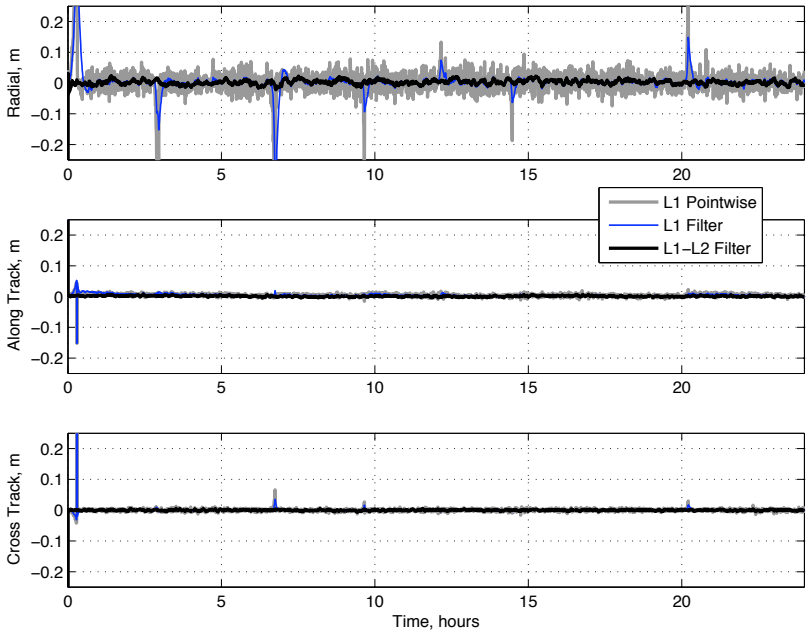

Figure 3.10: Relative position error components for a full orbit in GEO.

distance of 18 Earth radii and pass through perigee distance of 1.2 Earth radii about halfway through the scenario. It is assumed that the formation satellites have nadir-pointing GPS antennas with hemispheric gain patterns throughout the orbit. This orientation gives good GPS visibility when the formation navigates above the GPS constellation, but poor visibility when it navigates at lower altitudes, especially near perigee. This assumption, which may be reasonable from an operational standpoint, has the effect of testing an estimation algorithm's ability to navigate through perigee with as few as four GPS signals, all of which are influenced by the ionosphere. Figure 3.11 shows the three estimators' relative position error components in a particular full orbit HEO scenario. Both of the single-frequency estimators diverge from the correct integer ambiguity estimates and experience large navigation errors as the formation approaches perigee at 18.8 hours. Here, the GPS satellite visibility drops briefly from 11 or 12 satellites to 4 satellites. About 40 minutes later, the number of tracked satel-

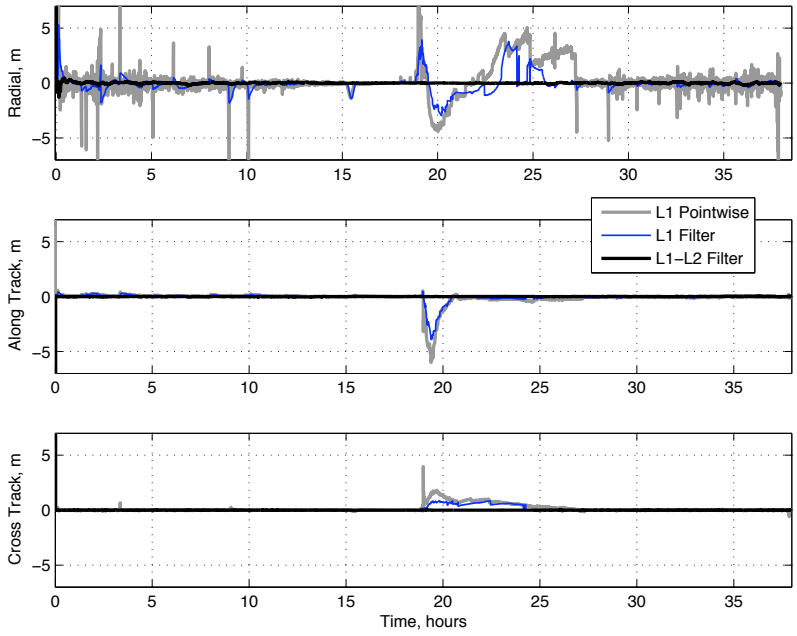

Figure 3.11: Relative position error components for a full orbit in HEO.

lites returns to 11 or 12, but the estimators do not recover the correct integers until about 25.5 hours in scenario time, or about 6 hours after the good visibility is restored. The dual-frequency filter, on the other hand, remains converged throughout the entire orbit and experiences navigation errors of less than 20 cm, excluding the initial transients. Figure 3.12 shows a zoomed in version of the same results.

3.7 Conclusions

A new dual-frequency CDGPS relative navigation filter has been developed for satellite formation flying at high-altitudes. It uses dynamics models for the spacecraft orbits, receiver clocks, ionospheric TEC, and GPS satellite residual position and clock errors. The orbital dynamics are modeled with a 2×2 grav-

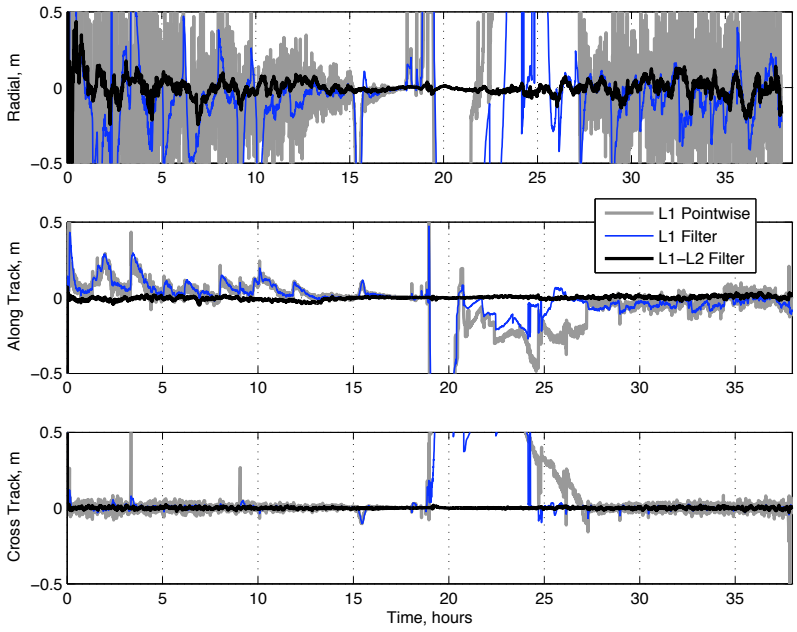

Figure 3.12: A zoomed in version of the relative position error components for a full orbit in HEO.

ity model with sun and moon perturbations. The spacecraft state process noise is divided into two parts, one that is common to both satellites in the formation and another that acts differentially and that is much smaller in magnitude. Each receiver clock is modeled as evolving according to a two-state random walk/drift process. Both the GPS satellite ephemeris errors and the ionospheric TEC are modeled as evolving according to first-order Markov processes. The TEC variables are split into an average TEC between the receivers, which is allowed to experience relatively large changes, and a differential TEC, which is adaptively tuned to identify those instances in which this quantity is significant and those in which it is negligible. To this end, the differential TEC noise intensity scaling varies as a function of the receiver-Earth-transmitter relative geometry. The filter tracks un-differenced carrier-phase ambiguities, thereby preserving information that is ordinarily discarded in CDGPS data processing,

but it still utilizes the integer nature of the double-differenced ambiguities in the relative navigation solution by applying a LAMBDA-type method after the measurement update.

The L1-L2 filter's performance in Monte Carlo simulations has been compared to two other similarly designed CDGPS algorithms, an L1-only pointwise estimator and an L1-only filter. The comparisons have focused on two template scenarios, a geostationary formation scenario and a high-Earth orbit scenario. The filters are far less noisy than the pointwise estimator in the absence of significant differential ionosphere effects. Averaged over quiescent periods in all Monte Carlo runs, the filters both had average error magnitudes of less than 1.2 cm in GEO, compared the pointwise estimator's 9.2 cm average. In HEO, the filters' mean error magnitudes were less than 13 cm, compared to the pointwise estimator's 43 cm. The relative navigation results are significantly improved with the addition of filtering.

When the differential ionosphere becomes significant, the L1-L2 filter outperformed the single-frequency estimators. During ionosphere events, the dual-frequency filter accurately estimated and removed the effect of the differential TEC and experienced no discernible degradation in navigation accuracy. The single-frequency estimators, on the other hand, experienced large errors during the same events. The L1-only filter error magnitude reached 4.41 m and the pointwise estimator's error magnitude reached 28.7 m.

The L1-L2 filter also exhibited better convergence and robustness qualities than did the L1-only estimators. It converged to the correct integer ambiguity estimates faster than the other estimators in every Monte Carlo run and never diverged from the correct integers. These improvements can be partially at-

tributed to the dual-frequency filter's ability to estimate and remove the effects of the differential ionosphere, but it also outperformed the other estimators in the absence of significant differential TEC. This latter behavior suggests that the L1-L2 filter can also use the dual-frequency measurements to aid in ambiguity resolution through implicit wide laning, and that, by considering the receiver-Earth-transmitter geometry, it is able choose how to best use the measurements on a channel-by-channel basis.

CHAPTER 4

CONTINUOUS-TIME KALMAN FILTERING WITH IMPLICIT DISCRETE MEASUREMENT TIMES

4.1 Abstract

A new approximation to a continuous-time Kalman filter has been developed, one that is useful for systems with implicitly defined measurement update times. This Kalman filter is applicable to radio navigation problems in which a state-dependent range delay must be determined by iterative solution of an implicit equation in order to compute the relevant Kalman filter measurement update time. The new filter consists of three main elements. The first element is a dense output numerical integration method that outputs a continuous description of the state over each integration interval. The second element is a new process noise model that approximates the underlying continuous-time white noise as a finite-order, piecewise polynomial. The third element is a new dynamic propagation/measurement update calculation that sensibly combines a dense output numerical integration scheme with the new process noise model and that estimates the process noise model's polynomial coefficients as discrete-time random variables such that the model optimally approximates continuous-time white noise. After developing the necessary theory, the method is demonstrated in simulation for an example tracking problem.

4.2 Introduction

Many radio navigation tracking systems make measurements at known receiver clock times. In some cases, the measurements are made onboard the vehicle whose state is to be estimated. An example of this type of system is the tracking of a satellite orbit based on data from an onboard GPS receiver. In that scenario, the state update due to the measurements occurs at an explicitly defined time. If the receiver is remote from the vehicle whose state is to be estimated, however, the navigation filter becomes more complicated. Consider the case illustrated in the upper panel of Fig. 4.1, where a ground station measures the pseudo-range using a downlink signal. In this scenario, the range component of the measurement represents the length of the signal's light path from the satellite's position at the signal transmit time to the ground station's position at the signal reception time. The true signal transmit time is unknown and must be determined iteratively during the Kalman filter's measurement update step. These iterations are represented in the figure by the multiple empty satellite images stepping backwards from time t to time $t - \delta t$, converging on the filled satellite. A state prediction (and covariance) must be computed for each iteration. A naive strategy for producing this prediction is to numerically re-propagate the state estimate (and covariance) from time t_k, which represents the time of the last state estimate, to each estimate of the time $t - \delta t$ that is generated for each new iteration. This approach quickly becomes computationally prohibitive, especially if the propagation algorithm evaluates detailed force models.[18] A second problem with this approach is that in certain cases the integration step size may become so small that roundoff errors become significant.[17]

A better approach is to employ a technique of state interpolation over pre-

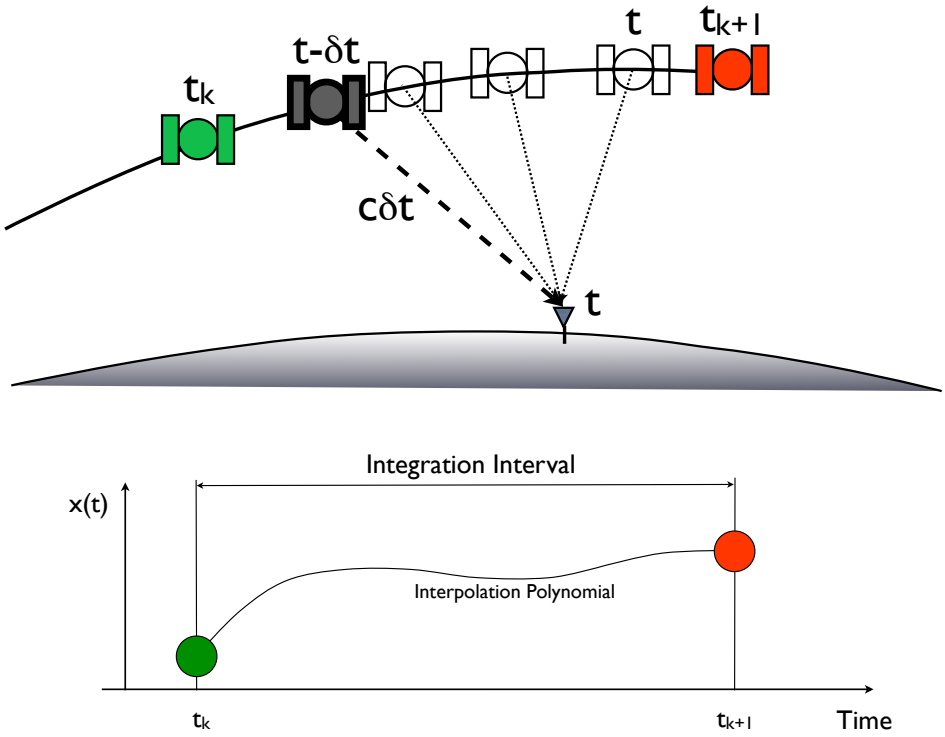

Figure 4.1: A typical downlink measurement scenario (upper panel) and a state interpolation scheme (lower panel). Scales are exaggerated.

determined integration intervals. Interpolated numerical integration, which is also referred to as dense output integration, allows the integration step size, e.g., from time t_k to t_{k+1}, to be determined independently of the measurement times by the user or by a step size control algorithm. The concept of state interpolation is represented graphically in the upper and lower panels of Fig. 4.1. In the upper panel, the satellite images at time t_k and time t_{k+1} represent the beginning and end of the predetermined integration interval. Corresponding satellite positions are represented in the lower panel by the circles. The curved line between those circles represents the interpolation polynomial that a dense output numerical

integration scheme would produce. Using this method, the predicted state can be evaluated anywhere over the interval during the signal transmit time iterations at the relatively cheap cost of a polynomial evaluation. The approach also has the benefit of completely decoupling the navigation filter's state estimate times from the measurement times.[17]

Typical orbit determination Kalman filters use dense output strategies similar to the one illustrated in Fig. 4.2.[17] The upper panel shows a single dense output integration step that begins at the initial state estimate (open circle), propagates the state over a fixed interval, and generates a state prediction polynomial (curved line). Two events happen over this interval. First, a user-defined output time t_k is crossed. Since no measurements have been encountered since the last a posteriori state estimate, the filter's best estimate of the state at time t_k is computed by evaluating the state prediction polynomial at that time (triangle), which is output to the user. Second, a measurement (filled circle) that was made at the signal reception time t is encountered. In order to generate the state prediction required to properly process the measurement, the filter evaluates the state prediction polynomial at the signal reception time t. This time, however, is not the correct time at which to apply the measurement, but is delayed by the signal's propagation time δt. The state prediction, therefore, is treated as the first in a series of candidate predictions (empty squares in the bottom panel) that converge on the correct prediction (filled square in the bottom panel) as the filter solves an implicit time of transmission equation iteratively as was illustrated in the upper panel of Fig. 4.1. Once a converged state prediction is found, a new state estimate is generated (empty circle in the bottom panel), and the numerical integration routine is restarted to generate a new trajectory polynomial. The process is then repeated (upper panel of Fig. 4.3).

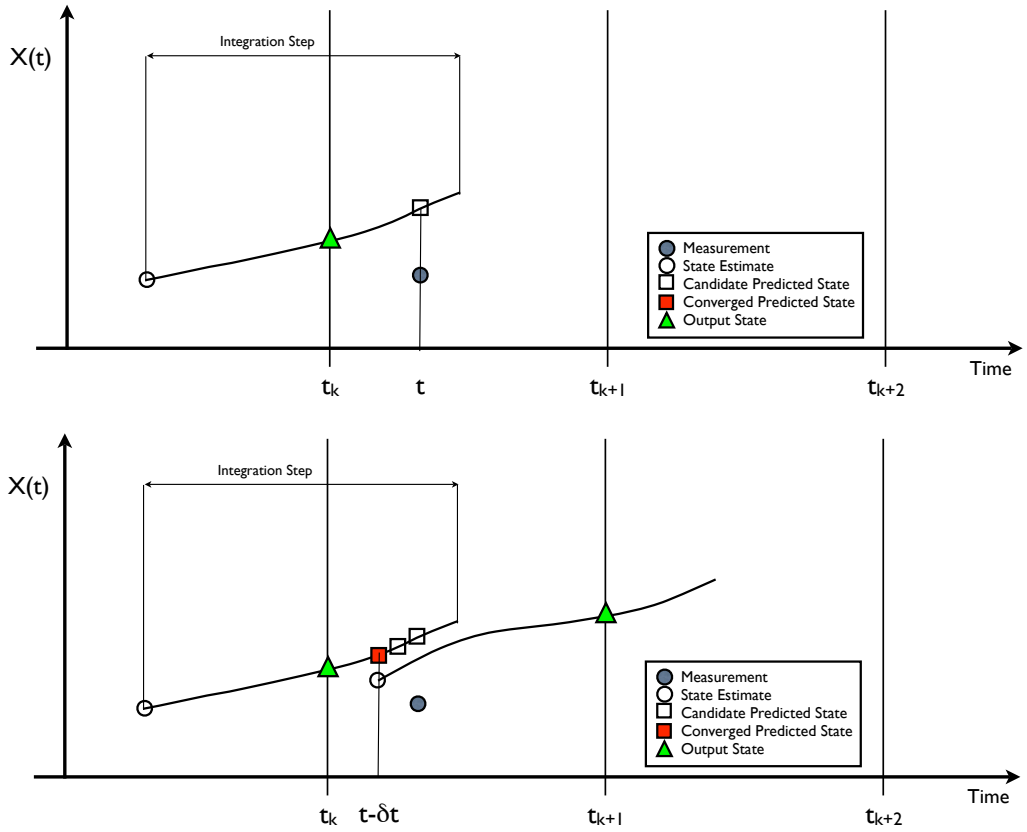

Figure 4.2: An extended Kalman filtering strategy that makes use of dense output numerical integration over a fixed interval. (Upper panel: prediction. Lower panel: measurement update and next prediction.)

Although this approach is widely implemented, it is not free of problems, particularly when the measurement density is high. First, and rather obviously, the restarting of the integration algorithm at each measurement time wastes computation by overlapping the integration intervals. A second, and more subtle, problem involves the Kalman filter's statistical model of the process noise and becomes important if the user is interested in obtaining realistic estimates of that noise, as in a smoother. It is common in sampled data filtering to assume a constant random process noise between measurement sample times. In the scheme depicted in Fig. 4.3, however, this would result in process noise discon-

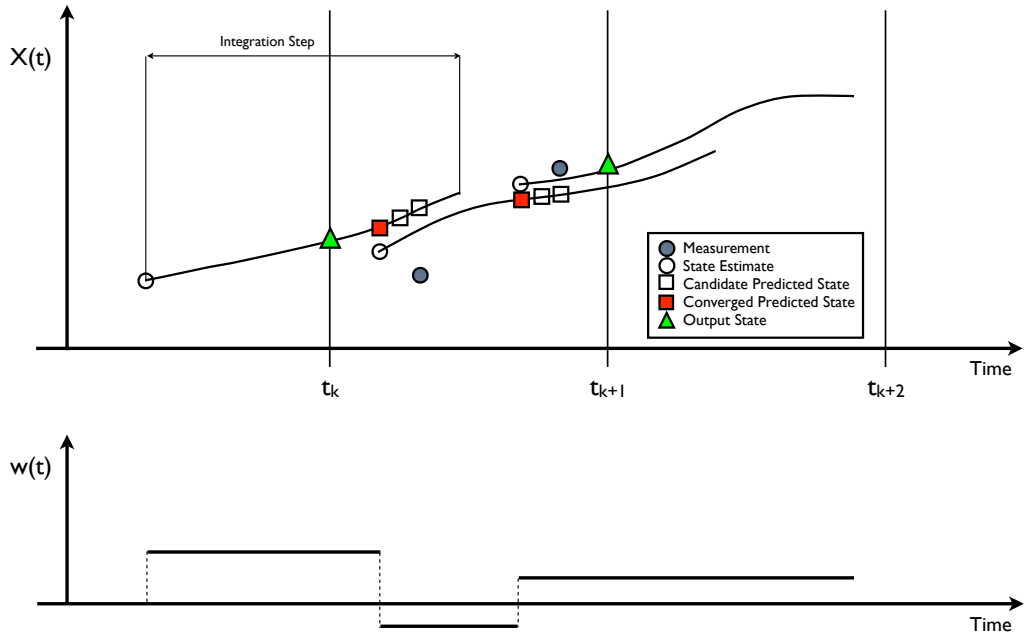

Figure 4.3: Multiple extended Kalman filter predictions and measurement updates (upper panel), and the traditional zero-order hold process noise model (lower panel).

tinuities within a dense output numerical integration interval. This situation is depicted in the zero-order hold process noise time history in the bottom panel of the figure. Such discontinuities would fundamentally violate the assumptions of the dense output numerical scheme.

Another difficulty with such schemes is that the measurement update implies a change in the measurement effectiveness time. This change occurs because the state estimate enters the equation that must be solved iteratively for the measurement effectiveness time. Note that this effect is neglected in Figs. 4.2 and 4.3 because it is difficult to depict properly.

This paper develops a different approach, illustrated in Fig. 4.4, in which multiple measurements taken over a given integration interval are processed all at once in a manner consistent with estimating the parameters of a process noise model that is continuous over the same interval. In the new approach, the output times of the filter coincide with the integration periods; the a posteriori state estimate is calculated only at the end of a given integration interval. Dense output state polynomials are used to generate state predictions during the measurement update signal transmit time iterations. The key to making this approach work is to derive a process noise model that maintains continuity and smoothness over an interval while approximating the statistics of white noise as best as possible. A continuous-time polynomial description of the process noise over each integration interval is proposed, as depicted in the bottom panel of Fig. 4.4. The polynomial coefficients are modeled as discrete-time random variables drawn from zero-mean white-noise sequences in a way that allows their associated polynomials to optimally approximates continuous-time white noise. This update procedure causes discontinuous transitions only at the state output times t_k, t_{k+1}, etc., and of necessity, it involves the estimation of the process noise polynomial coefficients in a partial smoothing step.

This paper makes three contributions. First, a nonlinear state trajectory model is developed that includes the polynomial process noise model and dense output numerical integration. Derivations are presented for all of the partial derivatives that relate both the final state and the measurements that are applicable at implicitly defined intermediate times to the initial state and the process noise polynomial coefficients. The linearized system is then assembled into a square-root information filter form, which is convenient for performing the state update while simultaneously estimating the process noise coefficients.[2]

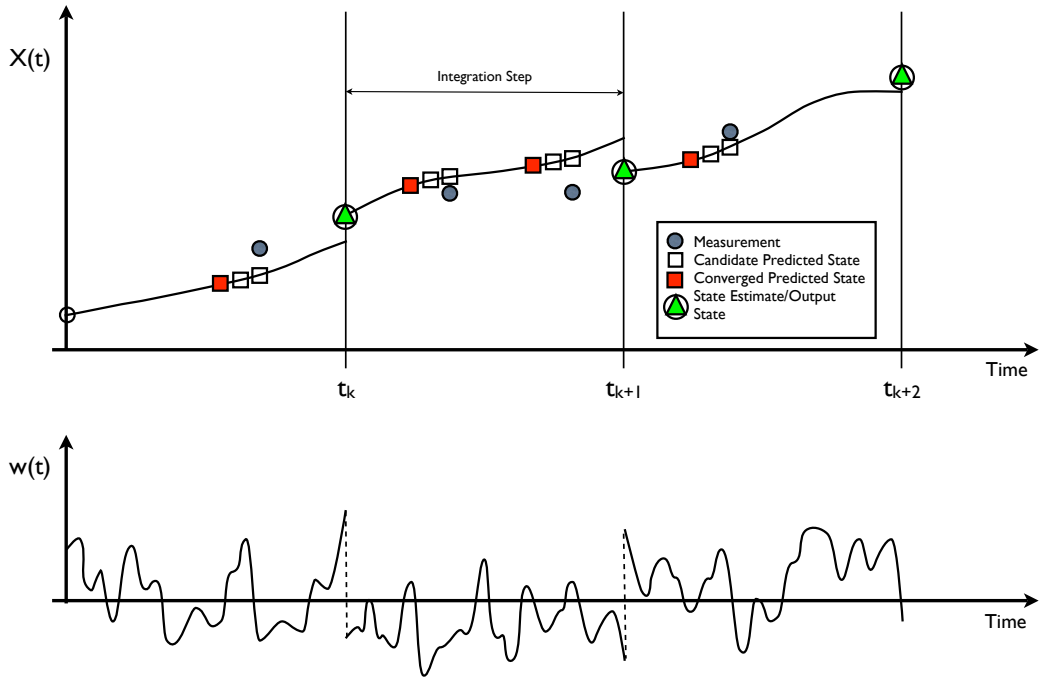

Figure 4.4: A modified extended Kalman filtering strategy for processing multiple, iterated measurements over a given integration interval (upper panel), and a piecewise polynomial process noise model (lower panel).

The second contribution is the development of an optimized statistical model of the polynomial process noise coefficients. This optimization causes the piecewise polynomial process noise to approximate continuous-time white noise as closely as possible. The third contribution uses the first two developments to design an extended Kalman filter that rationally handles implicitly defined measurement effectiveness times. Although this filtering technique is applicable to a variety of estimation problems, tracking problems with radio navigation measurements are used as examples throughout the paper. The validity of the new approach is tested in truth-model simulations using a simple tracking problem.

The remainder of the paper is divided into four major sections. Section II presents models for the state dynamics, the process noise, and the measurements, linearizes the resulting models, and discusses dense output numerical integration. Section II also incorporates the models into dynamic propagation and state update equations that deal with implicit measurement update times. Section III presents the statistical description of the piecewise polynomial process noise model and an optimization that causes it to approximate white noise. Section IV presents the tracking problem truth-model simulations and the results. Section V presents the conclusions.

4.3 Modeling and Linearization with Sampled Measurements, Implicit Update Times, and Dense Output Numerical Integration

A nonlinear state trajectory model that treats the process noise as a continuous-time polynomial over an integration interval is presented in this section. The relationships between the observables and the independent variables are linearized to obtain the simplified expressions that are needed to solve the nonlinear filtering problem using the principles of the extended Kalman filter. The resulting expressions are assembled into a non-standard form of a square-root information filter, one that performs its dynamic propagation and measurement update simultaneously.

4.3.1 Dynamics Modeling and Linearization with Dense Output

The trajectory model is expressed by the following nonlinear, continuous-time, ordinary differential equation with forcing:

$$\dot{\mathbf{x}}(t) = \mathbf{f}[t, \mathbf{x}(t), \mathbf{w}(t)] \tag{4.1}$$

where $\mathbf{x}(t)$ is the system's state vector at time t, $\mathbf{w}(t)$ is the process noise vector, and the function $\mathbf{f}$ models the system's dynamics and how the process noise influences them. Since the signal transmit times must be solved iteratively during the filter's measurement update step, it is desirable to have a polynomial description of the state trajectory over the integration interval from time t_k to t_{k+1}. A generic N^{th}-order model of such a polynomial takes the form

$$\mathbf{x}(t) = \sum_{i=0}^{N} \tilde{\mathbf{x}}_k^i T_i(t) \qquad for \quad t_k \leq t < t_{k+1} \tag{4.2}$$

where $\tilde{\mathbf{x}}_k^i$ is the i^{th} polynomial coefficient vector that is valid over the k^{th} integration interval, and where the function $T_i(t)$ is the i^{th} polynomial basis function defined over the interval $t_k \leq t < t_{k+1}$. It should be noted that there are many methods for developing such polynomials, including dense output Runge-Kutta methods and multi-step Adams-Bashforth methods.[10,18] Regardless of the method, an important consideration is that the resulting polynomials satisfy the order condition of the numerical method.[10] Runge-Kutta methods, which do not require the storage of states or function evaluations prior to the beginning of the current integration interval, are the most appropriate methods for use in sequential estimation and are considered exclusively in the remainder of the paper. For a more thorough discussion of continuous output numerical integration algorithms, please see Refs. 10 and 18.

The proposed process noise model over a particular integration interval takes a similar form:

$$\mathbf{w}_k(t) = \sum_{j=0}^{M} \tilde{\mathbf{w}}_k^j P_j(t) \qquad for \quad t_k \le t < t_{k+1} \tag{4.3}$$

where $\tilde{\mathbf{w}}_k^j$ is the j^{th} polynomial coefficient vector that is valid over the k^{th} integration interval, and the function $P_j(t)$ is the j^{th} basis function of an M^{th}-order generic polynomial. The choice of the particular polynomial basis functions used in Eq. (4.3) will almost certainly differ from ones used for the state propagation, as discussed in the next section, hence the differing letters used to designate them. The polynomial coefficient vectors are assumed to be constant over the integration interval and assumed to be drawn from white-noise sequences, which implies that $E[(\tilde{\mathbf{w}}_k^j)(\tilde{\mathbf{w}}_\ell^i)^T] = 0$ for all ji pairs when $k \neq \ell$, i.e., the process noise coefficients are uncorrelated in time.

Before continuing with the derivation of the model, it is important to point out the functional relationship between the state and the process noise coefficients explicitly. Although the $\tilde{\mathbf{w}}_k^j$ coefficients cannot be exactly known a priori, the derivation of the filter equations requires the development of a model of how they affect the state trajectory over the integration interval. With this in mind, the system's state over the interval from t_k to t_{k+1} has the following functional dependence:

$$\mathbf{x}(t; \mathbf{x}_k, t_k, t_{k+1}, \tilde{\mathbf{w}}_k^0, \tilde{\mathbf{w}}_k^1, \ldots, \tilde{\mathbf{w}}_k^M) \tag{4.4}$$

This dependence is central to the derivation of the linearized expressions presented later in this section. An algorithm for the explicit computation of this function can be developed by using the dynamics model in Eq. (4.1), the process noise model in Eq. (4.3), and any dense output numerical integration method. The Appendix presents one appropriate numerical method, a 5^{th}-order dense

output Runge-Kutta method that uses a 4^{th}-order interpolant. In order to simplify the notation, the set $\Sigma_k = \{\mathbf{x}_k, t_k, t_{k+1}, \tilde{\mathbf{w}}_k^0, \tilde{\mathbf{w}}_k^1, \ldots, \tilde{\mathbf{w}}_k^M\}$ is used to denote the interval-related dependencies in the subsequent expressions.

In order to produce state and process noise coefficient estimates, a filter requires a description of how changes in the initial state and process noise coefficients influence the state at any other time. This information is provided in the form of partial derivatives of the function in Eq. (4.4) with respect to its initial state and process noise coefficients. These relationships are derived in the next paragraphs.

Given the dynamics model in Eq. (4.1), the state trajectory over the interval from t_k to t_{k+1} depends on the initial state and on the process noise coefficients over the interval. Any changes in those values will result in changes in the state at time t_{k+1}. Consider first how changes in the initial state influence the final state. The partial derivative of the state at time t_{k+1} with respect to the state at time t_k describes this relationship and defines the state transition matrix as

$$\Phi_k = \frac{\partial \mathbf{x}(t_{k+1}; \Sigma_k)}{\partial \mathbf{x}_k} \tag{4.5}$$

For a general nonlinear model, no analytical expression for this matrix exists. It is computed by numerically integrating the following matrix differential equation from time t_k to t_{k+1}:

$$\dot{\Phi}(t, t_k) = A(t)\Phi(t, t_k) \tag{4.6}$$

starting from the initial condition $\Phi(t_k, t_k) = I$. In Eq. (4.6), the matrix $A(t)$ is defined as the partial derivative of the dynamics model with respect to the state:

$$A(t) = \frac{\partial \mathbf{f}[t, \mathbf{x}(t; \Sigma_k), \mathbf{w}(t)]}{\partial \mathbf{x}(t; \Sigma_k)} \tag{4.7}$$

This partial derivative can, in most cases, be evaluated analytically, although in

practice it is often evaluated using simplified models or finite difference methods.[18] It should be noted that the numerical integration of Eq. (4.6) can be performed at the same time as the numerical integration of the state vector and that dense output Runge-Kutta polynomials can be generated for the elements of the state transition matrix.

Now consider how changes in the process noise polynomial coefficients influence the state at time t_{k+1}. This sensitivity is expressed in the following process noise coefficient influence matrix for the j^{th} process noise coefficient:

$$\Gamma_k^j = \frac{\partial \mathbf{x}(t_{k+1}; \Sigma_k)}{\partial \tilde{\mathbf{w}}_k^j} \tag{4.8}$$

As with the state transition matrix, an analytic expression for this matrix normally does not exist. It is computed by numerically integrating the following matrix differential equation from time t_k to t_{k+1}:

$$\dot{\Gamma}^j(t, t_k) = A(t)\Gamma^j(t, t_k) + D_k^j(t) \tag{4.9}$$

starting from the initial condition $\Gamma^j(t_k, t_k) = 0$. In Eq. (4.9), the matrix $D_k^j(t)$ is defined as the partial derivative of the model with respect to the j^{th} process noise coefficient on the k^{th} integration interval:

$$D_k^j(t) = \frac{\partial \mathbf{f}[t, \mathbf{x}(t; \Sigma_k), \mathbf{w}(t)]}{\partial \mathbf{w}(t)} \frac{\partial \mathbf{w}(t)}{\partial \tilde{\mathbf{w}}_k^j} \tag{4.10}$$

The first partial derivative on the right-hand side of this equation is defined as $D(t) = \frac{\partial \mathbf{f}[t, \mathbf{x}(t; \Sigma_k), \mathbf{w}(t)]}{\partial \mathbf{w}(t)}$, and the second partial derivative on the right-hand side, given the definition in Eq. (4.3) of the continuous-time process noise vector over the k^{th} integration interval, evaluates to $P_j(t)$. Equation (4.10) thus simplifies to

$$D_k^j(t) = D(t)P_j(t) \tag{4.11}$$

In order to assemble these relationships into the linearized trajectory model, a vector of stacked process noise polynomial coefficients is defined as

$$\tilde{\mathbf{w}}_k = \begin{bmatrix} \tilde{\mathbf{w}}_k^0 \\ \tilde{\mathbf{w}}_k^1 \\ \vdots \\ \tilde{\mathbf{w}}_k^M \end{bmatrix} \tag{4.12}$$

and the associated process noise coefficient influence matrix is defined as

$$\Gamma_k = [\Gamma_k^0 \quad \Gamma_k^1 \quad \dots \quad \Gamma_k^M] \tag{4.13}$$

Defining a state deviation vector as $\Delta\mathbf{x}_k = \mathbf{x}(t_k; \Sigma_k) - \hat{\mathbf{x}}_k$, where $\hat{\mathbf{x}}_k$ is the a posteriori estimate of the state at time t_k, and assuming that $E[\tilde{\mathbf{w}}_k] = 0$ and that the model is linearized about the values $\mathbf{x}_k = \hat{\mathbf{x}}_k$ and $\tilde{\mathbf{w}}_k = 0$, the linearized trajectory model takes the form

$$\Delta\mathbf{x}_{k+1} = \Phi_k \Delta\mathbf{x}_k + \Gamma_k \tilde{\mathbf{w}}_k \tag{4.14}$$

where $\Delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}$. The term $\bar{\mathbf{x}}_{k+1}$ is the a priori estimate of $\mathbf{x}_{k+1}$ that results from propagating the differential equation from the initial condition $\mathbf{x}(t_k) = \hat{\mathbf{x}}_k$ to the terminal time t_{k+1} under the assumption that $\mathbf{w}(t) = 0$. Note that with dense output polynomials for the elements of $\Phi(t, t_k)$ and

$$\Gamma(t, t_k) = [\Gamma^0(t, t_k) \quad \Gamma^1(t, t_k) \quad \dots \quad \Gamma^M(t, t_k)] \tag{4.15}$$

it is also possible to write the state deviation equation at any point in the interval as a function of $\Delta\mathbf{x}_k$ and $\tilde{\mathbf{w}}_k$,

$$\Delta\mathbf{x}(t) = \Phi(t, t_k)\Delta\mathbf{x}_k + \Gamma(t, t_k)\tilde{\mathbf{w}}_k \tag{4.16}$$

Before developing the measurement model, it is useful to take a closer look at Eq. (4.16) in order to understand how the new polynomial process noise

model is related to the traditional zero-order hold model. Rewriting Eq. (4.16) such that the integrals used to define the large process noise influence matrix in Eq. (4.13) appear explicitly, the expression for the M^{th}-order polynomial process noise model is

$$\Delta\mathbf{x}(t) = \Phi(t,t_k)\Delta\mathbf{x}_k + \left(\int_{t_k}^{t} \Phi(t,\tau)D(\tau)\left[P_0(\tau)I \quad P_1(\tau)I \quad \ldots \quad P_M(\tau)I\right]d\tau\right)\begin{bmatrix}\tilde{\mathbf{w}}_k^0\\ \tilde{\mathbf{w}}_k^1\\ \vdots\\ \tilde{\mathbf{w}}_k^M\end{bmatrix} \tag{4.17}$$

In this form, it is easy to see how the new model is related to the zero-order hold model. Consider the case where the first process noise coefficient, $\tilde{\mathbf{w}}_k^0$, is non-zero and all the others are zero, and where the zeroth-order basis function, $P_0(\tau)$, is constant and equal to one. Equation (4.17) collapses from the M^{th}-order model to the zeroth-order model, which is identical to the zero-order hold model. As successive polynomial coefficients take on non-zero values, the zero-order hold process noise model is augmented with a linear term, then a quadratic term, and so on, up to the M^{th}-order term. The M^{th}-order polynomial process noise model, thus, represents an extension of the zero-order hold model rather than a departure from it.

4.3.2 Measurement Modeling with Implicit Measurement Times

A general, nonlinear, implicitly constrained measurement model takes the form

$$\mathbf{y}_{k(q)} = \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)] + \mathbf{v}_{k(q)} \tag{4.18}$$

subject to the constraint

$$0 = g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)] \tag{4.19}$$

where $\bar{t}_{k(q)} = t_{k(q)} - \delta t_{k(q)}$ with $t_{k(q)}$ being a known signal reception time and $\delta t_{k(q)}$ being an unknown transmission delay. In Eq. (4.18), $\mathbf{y}_{k(q)}$ is a discrete-time vector measurement that contains information about the state at the signal transmit time, $\bar{t}_{k(q)}$. The subscript $k(q)$ indicates the q^{th} measurement that was received on the k^{th} interval, i.e., the interval from t_k to t_{k+1}. The function $\mathbf{h}_{k(q)}$ models the nonlinear relationship between the state and the measurement vector. Equation (4.18) includes a measurement noise vector, $\mathbf{v}_{k(q)}$, which is assumed to be drawn from a zero-mean white-noise sequence and to be uncorrelated to the process noise. For a downlink radio navigation example, the function $g_{k(q)}$ of Eq. (4.19) constrains the speed of light multiplied by the propagation time, $c\delta t_{k(q)}$, to equal the range between the radio navigation receiver's position at reception time, $t_{k(q)}$, and the vehicle's position at the signal transmit time, $\bar{t}_{k(q)}$. The transmit time, which is defined in Eq. (4.19) as an implicit function of Σ_k, is determined by solving Eq. (4.19) using any appropriate numerical equation solving technique.

For multiple measurements on a given interval, i.e., for $q = 1, 2, \ldots$, there are multiple versions of Eq. (4.19) that must be solved for multiple transmission times, $\bar{t}_{k(q)}$. Once the signal transmit times have been determined, these multiple measurements can be stacked into a single large measurement vector that applies for the entire interval.

Given this measurement model with the implicit measurement times, the extended Kalman filter requires Jacobian partial derivatives of the model in order to carry out its measurement update calculation. These include partial derivatives of $\mathbf{h}_{k(q)}$ with respect to $\mathbf{x}_k$ and $\tilde{\mathbf{w}}_k$. The correct formulas must properly ac-

count for the dependence of $\mathbf{x}(t;\Sigma_k)$ on $\mathbf{x}_k$ and $\tilde{\mathbf{w}}_k$ and for the dependence of $\bar{t}_{k(q)}$ on these quantities. Starting with the partial derivatve with respect to $\mathbf{x}_k$, all the possible dependency paths from the initial state to the measurements are accounted for through the application of the chain rule. The following expression is derived:

$$H_{\mathbf{x}_{k(q)}} = \frac{\partial \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)} \left[\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \mathbf{x}_k} + \frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k} \right] + \frac{\partial \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k} \quad (4.20)$$

The first partial derivative on the right-hand side of Eq. (4.20) indicates how changes in the state at the signal transmit time, $\bar{t}_{k(q)}$, influence the measurement. An analytical expression for this partial derivative is easy to derive for most measurement models. The state at the signal transmit time, in turn, is influenced by changes in the initial state, $\mathbf{x}_k$, in two different ways, which are indicated by the bracketed partial derivatives. First, changes in the initial state influence the state at time $\bar{t}_{k(q)}$ directly through the partial derivative $\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \mathbf{x}_k}$, which is recognized as the state transition matrix from time t_k to $\bar{t}_{k(q)}$. This partial derivative may be obtained by evaluating the Runge-Kutta method's dense output polynomials. The second dependency path is through the signal transmit time variable. Changes in the signal transmit time influence the state at the transmit time directly through partial derivative $\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \bar{t}_{k(q)}}$, which is recognized as the time derivative of the state.* The signal transmit time is influenced by changes in the initial state as expressed by the partial derivative $\frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k}$. The measurement may also be influenced directly by changes in the signal transmit time. That dependence is represented in the first partial derivative outside the

*This time derivative normally comes from the dynamics model in Eq. (4.1), though there might be reason to use the explicit time derivative of the dense output polynomial in Eq. (4.2). These will be close to each other by the construction of the dense output numerical scheme, but probably not equal.

brackets, $\frac{\partial \mathbf{h}_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \bar{t}_{k(q)}}$. The last term in Eq. (4.20), $\frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k}$, represents, again, the dependence of the signal transmit time on the changes in the initial state.

In order to evaluate this last term, $\frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k}$, one takes the partial derivative of the constraint in Eq. (4.19) with respect to $\mathbf{x}_k$. Using the chain rule, this partial derivative is

$$0 = \frac{\partial g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)} \left[\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \mathbf{x}_k} + \frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k} \right] + \frac{\partial g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k} \quad (4.21)$$

The partial derivatives in Eq. (4.21) indicate that changes in the initial state, $\mathbf{x}_k$, influence the constraint in the same way as they influence the measurement. Note the similarity between Eq. (4.20) and Eq. (4.21). The important point when considering Eq. (4.21) is that the partial derivative, $\frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k}$, enters linearly, and the solution to this linear equation is

$$\frac{\partial \bar{t}_{k(q)}}{\partial \mathbf{x}_k} = \frac{\left[-\frac{\partial g_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)} \frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \mathbf{x}_k} \right]}{\left[\frac{\partial g_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)} \frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \bar{t}_{k(q)}} + \frac{\partial g_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \bar{t}_{k(q)}} \right]} \quad (4.22)$$

It includes a combination of partial derivatives that can be evaluated analytically, e.g., $\frac{\partial g_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}$ and $\frac{\partial g_{k(q)}[t_{k(q)},\bar{t}_{k(q)},\mathbf{x}(\bar{t}_{k(q)};\Sigma_k)]}{\partial \bar{t}_{k(q)}}$, and partial derivatives that have already been discussed, e.g., the state transition matrix from time t_k to $\bar{t}_{k(q)}$, $\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \mathbf{x}_k}$, and the time derivative of the state at time $\bar{t}_{k(q)}$, $\frac{\partial \mathbf{x}(\bar{t}_{k(q)};\Sigma_k)}{\partial \bar{t}_{k(q)}}$.

Next, the partial derivative of a measurement with respect to the process noise polynomial coefficients is derived in a completely analogous way. The results for the measurement's dependence on the j^{th} polynomial coefficient are

presented in the next two equations:

$$H^{j}_{\tilde{\mathbf{w}}_{k(q)}} = \frac{\partial \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)} \left[\frac{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)}{\partial \tilde{\mathbf{w}}^{j}_{k}} + \frac{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \tilde{\mathbf{w}}^{j}_{k}} \right] + \frac{\partial \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)]}{\partial \bar{t}_{k(q)}} \frac{\partial \bar{t}_{k(q)}}{\partial \tilde{\mathbf{w}}^{j}_{k}} \quad (4.23)$$

$$\frac{\partial \bar{t}_{k(q)}}{\partial \tilde{\mathbf{w}}^{j}_{k}} = \frac{\left[-\frac{\partial g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)} \frac{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)}{\partial \tilde{\mathbf{w}}^{j}_{k}} \right]}{\left[\frac{\partial g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)]}{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)} \frac{\partial \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)}{\partial \bar{t}_{k(q)}} + \frac{\partial g_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \mathbf{x}(\bar{t}_{k(q)}; \Sigma_k)]}{\partial \bar{t}_{k(q)}} \right]} \quad (4.24)$$

Equations (4.23) and (4.24) are evaluated for each of the $M + 1$ process noise polynomial coefficients, and the results are assembled into the form

$$H_{\tilde{\mathbf{w}}_{k(q)}} = [H^{0}_{\tilde{\mathbf{w}}_{k(q)}} \quad H^{1}_{\tilde{\mathbf{w}}_{k(q)}} \quad \dots \quad H^{M}_{\tilde{\mathbf{w}}_{k(q)}}] \quad (4.25)$$

An observed-minus-modeled measurement vector is defined as $\Delta \mathbf{y}_{k(q)} = \mathbf{y}_{k(q)} - \mathbf{h}_{k(q)}[t_{k(q)}, \bar{t}_{k(q)}, \bar{\mathbf{x}}(\bar{t}_{k(q)}; \Sigma_k)]$, and the linearized measurement model is defined as

$$\Delta \mathbf{y}_{k(q)} = H_{\mathbf{x}_{k(q)}} \Delta \mathbf{x}_k + H_{\tilde{\mathbf{w}}_{k(q)}} \tilde{\mathbf{w}}_k + \mathbf{v}_{k(q)} \quad (4.26)$$

Stacking all of the measurements $q = 1, 2, \dots, N_{m_k}$ for the sample interval from t_k to t_{k+1} results in the following definitions:

$$\Delta \mathbf{y}_k = \begin{bmatrix} \Delta \mathbf{y}_{k(1)} \\ \Delta \mathbf{y}_{k(2)} \\ \vdots \\ \Delta \mathbf{y}_{k(N_{m_k})} \end{bmatrix}; \qquad H_{\mathbf{x}_k} = \begin{bmatrix} H_{\mathbf{x}_{k(1)}} \\ H_{\mathbf{x}_{k(2)}} \\ \vdots \\ H_{\mathbf{x}_{k(N_{m_k})}} \end{bmatrix}; \qquad H_{\tilde{\mathbf{w}}_k} = \begin{bmatrix} H_{\tilde{\mathbf{w}}_{k(1)}} \\ H_{\tilde{\mathbf{w}}_{k(2)}} \\ \vdots \\ H_{\tilde{\mathbf{w}}_{k(N_{m_k})}} \end{bmatrix} \quad (4.27)$$

and the final linearized measurement equation:

$$\Delta \mathbf{y}_k = H_{\mathbf{x}_k} \Delta \mathbf{x}_k + H_{\tilde{\mathbf{w}}_k} \tilde{\mathbf{w}}_k + \mathbf{v}_k \quad (4.28)$$

4.3.3 Combined SRIF Dynamics Propagation and Measurement Update

In order to solve the linearized estimation problem with a square-root information filter, in a generalization of the method in Ref. 2, the following information equations are constructed:

$$\hat{\mathbf{z}}_{\mathbf{x}_k} = \hat{R}_{xx_k}\Delta\mathbf{x}_k + \mathbf{v}_{\mathbf{x}_k} \tag{4.29}$$

and

$$\mathbf{z}_{\tilde{\mathbf{w}}_k} = R_{\tilde{w}\tilde{w}_k}\tilde{\mathbf{w}}_k + \mathbf{v}_{\tilde{\mathbf{w}}_k} \tag{4.30}$$

where the noise terms, $\mathbf{v}_{\mathbf{x}_k}$ and $\mathbf{v}_{\tilde{\mathbf{w}}_k}$, are independent, zero-mean, identity-covariance, white-noise random vectors. The matrices, $\hat{R}_{xx_k}$ and $R_{\tilde{w}\tilde{w}_k}$, are defined as the inverse square roots of the a posteriori state and a priori process noise coefficient covariances, e.g., $\hat{R}^{-1}_{xx_k}\hat{R}^{-T}_{xx_k} = P_{xx_k}$ and $R^{-1}_{\tilde{w}\tilde{w}_k}R^{-T}_{\tilde{w}\tilde{w}_k} = Q_k$. The information equations are then combined with the measurements in the vector-matrix equation

$$\begin{bmatrix} \mathbf{z}_{\tilde{\mathbf{w}}_k} \\ \hat{\mathbf{z}}_{\mathbf{x}_k} \\ \Delta\mathbf{y}_k \end{bmatrix} = \begin{bmatrix} R_{\tilde{w}\tilde{w}_k} & 0 \\ 0 & \hat{R}_{xx_k} \\ H_{\tilde{\mathbf{w}}_k} & H_{\mathbf{x}_k} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_k \\ \Delta\mathbf{x}_k \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{\tilde{\mathbf{w}}_k} \\ \mathbf{v}_{\mathbf{x}_k} \\ \mathbf{v}_k \end{bmatrix} \tag{4.31}$$

It is assumed that the measurements are transformed appropriately such that the measurement noise vector, $\mathbf{v}_k$, is a zero-mean, identity-covariance, white-noise random vector. The linearized trajectory model in Eq. (4.14) is used to eliminate $\Delta\mathbf{x}_k$ from Eq. (4.31) in order to yield the following vector-matrix equation:

$$\begin{bmatrix} \mathbf{z}_{\tilde{\mathbf{w}}_k} \\ \hat{\mathbf{z}}_{\mathbf{x}_k} \\ \Delta\mathbf{y}_k \end{bmatrix} = \begin{bmatrix} R_{\tilde{w}\tilde{w}_k} & 0 \\ -\hat{R}_{xx_k}\Phi_k^{-1}\Gamma_k & \hat{R}_{xx_k}\Phi_k^{-1} \\ \left[H_{\tilde{\mathbf{w}}_k} - H_{\mathbf{x}_k}\Phi_k^{-1}\Gamma_k\right] & H_{\mathbf{x}_k}\Phi_k^{-1} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_k \\ \Delta\mathbf{x}_{k+1} \end{bmatrix} + \begin{bmatrix} \mathbf{v}_{\tilde{\mathbf{w}}_k} \\ \mathbf{v}_{\mathbf{x}_k} \\ \mathbf{v}_k \end{bmatrix} \tag{4.32}$$

Next, the propagation and measurement update steps are performed simultaneously by orthogonal/upper triangular (QR) factorization according to standard square-root information filtering techniques, resulting in the following a posteriori system:

$$\begin{bmatrix} \hat{\mathbf{z}}_{\tilde{\mathbf{w}}_k} \\ \hat{\mathbf{z}}_{\mathbf{x}_{k+1}} \\ \mathbf{z}_{res} \end{bmatrix} = \begin{bmatrix} \hat{R}_{\tilde{w}\tilde{w}_k} & \hat{R}_{\tilde{w}x_{k+1}} \\ 0 & \hat{R}_{xx_{k+1}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{w}}_k \\ \Delta\mathbf{x}_{k+1} \end{bmatrix} + \begin{bmatrix} \hat{\mathbf{v}}_{\tilde{\mathbf{w}}_k} \\ \hat{\mathbf{v}}_{\mathbf{x}_{k+1}} \\ \mathbf{v}_{res} \end{bmatrix} \tag{4.33}$$

This equation may be used to generate the a posteriori state estimate, the process noise polynomial coefficient estimates, and the associated covariances according to the following formulas:

$$\Delta\hat{\mathbf{x}}_{k+1} = \hat{R}^{-1}_{xx_{k+1}} \hat{\mathbf{z}}_{\mathbf{x}_{k+1}} \tag{4.34}$$

$$\hat{\mathbf{x}}_{k+1} = \bar{\mathbf{x}}_{k+1} + \Delta\hat{\mathbf{x}}_{k+1} \tag{4.35}$$

$$P_{xx_{k+1}} = \hat{R}^{-1}_{xx_{k+1}} \hat{R}^{-T}_{xx_{k+1}} \tag{4.36}$$

$$\hat{\tilde{\mathbf{w}}}_k = \hat{R}^{-1}_{\tilde{w}\tilde{w}_k} [\hat{\mathbf{z}}_{\tilde{\mathbf{w}}_k} - \hat{R}_{\tilde{w}x_{k+1}} \Delta\hat{\mathbf{x}}_{k+1}] \tag{4.37}$$

$$P_{\tilde{w}\tilde{w}_k} = \hat{R}^{-1}_{\tilde{w}\tilde{w}_k} [I + (\hat{R}_{\tilde{w}x_{k+1}} \hat{R}^{-1}_{xx_{k+1}})(\hat{R}_{\tilde{w}x_{k+1}} \hat{R}^{-1}_{xx_{k+1}})^T] \hat{R}^{-T}_{\tilde{w}\tilde{w}_k} \tag{4.38}$$

An important point has been glossed over in the preceding. In the process noise coefficient information equation, Eq. (4.30), the matrix $R_{\tilde{w}\tilde{w}_k}$ that meets the requirement that the noise sequence $\mathbf{v}_{\tilde{\mathbf{w}}_k}$ is zero-mean and identity-covariance was assumed to be given. How is that matrix determined? Also, how does one choose $R_{\tilde{w}\tilde{w}_k}$ such that the continuous-time polynomial process noise is as white as possible over the integration interval? These questions are answered in the next section.

4.4 Optimal Approximation to Continuous-Time White Noise Via Random Piecewise Polynomials

This section extends the definition of the polynomial process noise model in Eq. (4.3) for all integration intervals and develops the model's statistics. That development is guided by the basic assumption of the Kalman filter that the system's state is a Markov process, i.e., that all of the information necessary to propagate the state is summed up in the state at a given time. This assumption relies on the complete unpredictability of the process noise due to its whiteness. If the process noise were autocorrelated, i.e., colored, the states prior to the given time would be needed to predict future process noise values, and thus future state values in some fashion.[1] Therefore, in order to satisfy the Markov process assumption, the system must be driven by white noise, which is usually also assumed to be zero-mean.[†] The statistical behavior of a generic continuous-time Gaussian white-noise process is completely characterized by its first two moments, i.e., its mean and covariance, as follows:

$$E[\mathbf{n}(t)] = 0 \tag{4.39}$$

$$E[\mathbf{n}(t)\mathbf{n}^T(\tau)] = Q\delta(t-\tau) \tag{4.40}$$

where Q is the (possibly time-varying) process noise intensity matrix, and where $\delta(t-\tau)$ is the Dirac delta function. The goal of this section is to develop the statistics of the proposed polynomial process noise model such that they approximate as closely as possible, under certain constraints, the standard white noise statistics.

[†]If this whiteness condition is not satisfied, the model can be augmented with a shaping filter that is driven by white process noise. If the process noise has a non-zero, but known, mean value, that value can be treated as a deterministic input.

4.4.1 Analysis and Design of Polynomial White Noise Approximation

The proposed process noise model is defined for all time by repeating Eq. (4.3) over all possible integration intervals, which are indicated by the index k. The resulting infinite sum defines a piecewise polynomial process noise model as follows:

$$\mathbf{w}(t) = \sum_{k=-\infty}^{\infty} \sum_{j=0}^{M} \tilde{\mathbf{w}}_k^j P_j\left(\frac{2}{\Delta t}[t - t_k] - 1\right) \tag{4.41}$$

where the j^{th} piecewise polynomial basis function is chosen to be

$$P_j(\eta) = \begin{cases} C_j(\eta) & if \quad -1 \le \eta < 1 \\ 0 & otherwise \end{cases} \tag{4.42}$$

and where $C_j(\eta)$ is some appropriate polynomial in η, such as a Chebyshev polynomial. Notice that the piecewise polynomial $P_j(\eta)$ is non-zero only when the normalized argument, η, is on the interval from -1 to 1.

Since this model includes discrete-time polynomial coefficients and a continuous-time description of process noise over each integration interval, a combined continuous/discrete-time approach is used to analyze its statistics with the goal of developing an analysis-based statistical design that causes Eq. (4.41) to approximate white noise.

Note that Δt is the length of a single integration interval. The following analysis assumes that $\Delta t = t_{k+1} - t_k$ is constant; that is, it is independent of k. This assumption simplifies the statistical analysis, but probably could be relaxed. The timeline illustrated in Fig. 4.5 shows the integration intervals for the range of k from -1 to 2. Note that the points $-\frac{\Delta t}{2}$, $\frac{\Delta t}{2}$, $\frac{3\Delta t}{2}$, and $\frac{5\Delta t}{2}$ are the midpoints of

the intervals over which the four sets of polynomials apply.

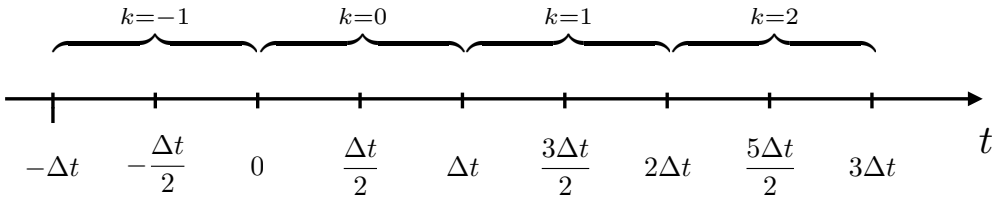

Figure 4.5: Timeline.

The key to developing the statistics of the piecewise polynomial process noise model lies in the modeling of the statistics of the polynomial's discrete-time coefficients. It is assumed that the coefficients are drawn from a zero-mean white-noise sequence. The mean for the j^{th} polynomial element coefficient is, thus,

$$E[\tilde{\mathbf{w}}_k^j] = 0 \tag{4.43}$$

and the covariance for the ji^{th} pair is

$$E[\tilde{\mathbf{w}}_k^j(\tilde{\mathbf{w}}_\ell^i)^T] = \alpha_{ji} Q \delta_{k\ell} \tag{4.44}$$

Here, α_{ji} is a scaling factor, Q is the continuous-time process noise intensity matrix, assumed constant in order to simplify the analysis, and $\delta_{k\ell}$ is the Kronecker delta function. The combined term $\alpha_{ji}Q$ represents the discrete-time process noise correlation for the ji^{th} pair with $\alpha_{jj}Q$ being the covariance of $\tilde{\mathbf{w}}_k^j$. The remainder of this section discusses the design of the α_{ji} scaling factors based on an analysis of how they affect the fidelity with which Eq. (4.41) approximates continuous-time white noise of intensity Q.

This analysis starts by considering the mean and autocorrelation of the model in Eq. (4.41). The mean is

$$E[\mathbf{w}(t)] = 0 \tag{4.45}$$

which follows from the assumption that the polynomial coefficients are drawn from a zero-mean white-noise sequence. The autocorrelation is

$$E[\mathbf{w}(t)\mathbf{w}^T(\tau)] = E\left[\left\{\sum_{k=-\infty}^{\infty}\sum_{j=0}^{M}\tilde{\mathbf{w}}_k^j P_j\left(\frac{2}{\Delta t}[t-t_k]-1\right)\right\}\left\{\sum_{\ell=-\infty}^{\infty}\sum_{i=0}^{M}\tilde{\mathbf{w}}_\ell^i P_i\left(\frac{2}{\Delta t}[\tau-t_l]-1\right)\right\}^T\right] \tag{4.46}$$

$$= Q\sum_{j,i=0}^{M}\alpha_{ji}R_{ji}(t-\tau) \tag{4.47}$$

where

$$R_{ji}(t-\tau) = E\left[\sum_{k=-\infty}^{\infty}P_j\left(\frac{2}{\Delta t}[t-t_k]-1\right)P_i\left(\frac{2}{\Delta t}[\tau-t_k]-1\right)\right] \tag{4.48}$$

$$= \begin{cases} \frac{1}{2}\int_{\gamma_{min}}^{\gamma_{max}} C_j(\frac{2}{\Delta t}[t-\tau]+\gamma)C_i(\gamma)d\gamma & |\tau-t_k|\le 1 \\ 0 & |\tau-t_k|>1 \end{cases} \tag{4.49}$$

and where

$$\gamma_{min} = -\frac{2}{\Delta t}min[0, t-\tau]-1 \tag{4.50}$$

$$\gamma_{max} = -\frac{2}{\Delta t}max[0, t-\tau]+1 \tag{4.51}$$

The derivation of Eq. (4.49) uses the principle of ergodicity by which the expectation value in Eq. (4.48) can be computed using a time average. This time average has been computed by integrating each term in the summation with respect to γ_k using the change of variable $\gamma_k = \frac{2}{\Delta t}(\tau - t_k) - 1$. The substitution $t_k = \tau - (\gamma_k + 1)\frac{\Delta t}{2}$ allows one to recognize that all of the integrals become identical for all of the different values of k; therefore, the subscript k can be dropped

from the γ_k dummy integration variable. This calculation also recognizes that the piecewise polynomials in Eq. (4.48) go to zero when their arguments have absolute values greater than one. As a consequence, $R_{ji}(t-\tau)$ is zero if $|t-\tau| \geq \Delta t$, consistent with the fact that $\gamma_{min} \leq \gamma_{max}$ only if $|t-\tau| \leq \Delta t$. Thus, the integral in Eq. (4.49) only needs to be evaluated when $\gamma_{min} < \gamma_{max}$.

Consistent with the form of $R_{ji}(t-\tau)$, the right-hand side of Eq. (4.49) depends only on $t-\tau$, which enters both through its integrand and through its limits of integration. This dependence ensures that the right-hand side of Eq. (4.47) has a form similar to the desired form on the right-hand side of Eq. (4.40). If the summation on the right-hand side of Eq. (4.47) were equal to a Dirac delta function, then this process noise model would be white noise.

With the proposed process noise model's autocorrelation in the form defined by Eqs. (4.47) and (4.49), it is now possible to develop a method for choosing the scaling factors α_{ji} for all of the coefficient pairs. The goal is to design α_{ji} scaling factors such that the summation in Eq. (4.47) approximates the Dirac delta function in Eq. (4.40) as closely as possible. Notice that cross- and autocorrelation functions between the polynomial elements are completely determined by the specific polynomials used, and that for each set of polynomials, a corresponding set of tuning parameters can be chosen such that the polynomial process noise approximates white noise as closely as possible. Before those scaling factors are chosen, however, there is another constraint to consider that only becomes clear when the joint statistics of all of the polynomial coefficients are considered.

Using the large vector of stacked polynomial coefficients defined in Eq. (4.12), and recalling that these values are assumed to be zero-mean, the process

noise coefficient covariance matrix is

$$Q_k = E[\tilde{\mathbf{w}}_k \tilde{\mathbf{w}}_k^T] = \begin{bmatrix} \alpha_{00}Q & \alpha_{10}Q & \dots & \alpha_{M0}Q \\ \alpha_{10}Q & \alpha_{11}Q & & \vdots \\ \vdots & & \ddots & \\ \alpha_{M0}Q & \dots & & \alpha_{MM}Q \end{bmatrix} \tag{4.52}$$

This matrix is constrained to be symmetric and positive semidefinite. It can be easily shown that this constraint requires the symmetric matrix

$$A_\alpha = \begin{bmatrix} \alpha_{00} & \alpha_{10} & \dots & \alpha_{M0} \\ \alpha_{10} & \alpha_{11} & & \vdots \\ \vdots & & \ddots & \\ \alpha_{M0} & \dots & & \alpha_{MM} \end{bmatrix} \tag{4.53}$$

to be positive semidefinite as well.

The choice of the scaling factors α_{ji} can be posed as a nonlinear, constrained optimization problem. Several strategies for solving this problem have been considered, including methods based on semidefinite programming and methods based on standard nonlinear optimization techniques combined with factorization techniques that ensure the positive semidefiniteness of A_α. For simplicity, however, this study has limited the optimization to the diagonal elements of the A_α matrix, setting all of the off-diagonal elements to be zero. This approach, which reduces the optimization to a quadratic programming problem, admits a simple form of the positive semidefinite constraint, but still provides the optimization enough flexibility to choose α_{jj} coefficients that yield a good approximation to white noise. A further study may reveal that optimizing over all of the elements in A_α is worthwhile.

The posing of a sensible quadratic programming problem begins by defining a new function that is the sum of the products of the scaling factors and the

polynomial autocorrelation functions:

$$B(\tau) = \sum_{j=0}^{M} \alpha_{jj} R_{jj}(\tau) \tag{4.54}$$

If the autocorrelation functions R_{jj} for all j are uniformly sampled N_s times over the interval from $-\Delta t$ to Δt, a sampled version of $B(\tau)$ can be constructed. It takes the following vector form:

$$\mathbf{b} = [B(\tau_1) \quad B(\tau_2) \quad \dots \quad B(\tau_{N_s})]^T \tag{4.55}$$

$$= R_s \boldsymbol{\alpha} \tag{4.56}$$

where the following definitions apply:

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_{00} \\ \alpha_{11} \\ \vdots \\ \alpha_{MM} \end{bmatrix} \tag{4.57}$$

is an $(M+1) \times 1$ vector and

$$R_s = \begin{bmatrix} R_{00}(\tau_1) & R_{11}(\tau_1) & \dots & R_{MM}(\tau_1) \\ R_{00}(\tau_2) & R_{11}(\tau_2) & \dots & R_{MM}(\tau_2) \\ \vdots & \vdots & & \vdots \\ R_{00}(\tau_{N_s}) & R_{11}(\tau_{N_s}) & \dots & R_{MM}(\tau_{N_s}) \end{bmatrix} \tag{4.58}$$

is an $N_s \times (M+1)$ matrix. In Eqs. (4.55) and (4.58), the i^{th} sample time is defined as $\tau_i = \frac{2\Delta t}{N_s}(i - \frac{1}{2}) - \Delta t$ for $i = 1, 2, \dots, N_s$.

The vector of diagonal scaling factors $\boldsymbol{\alpha}$ is chosen by solving the following quadratic programming problem: minimize the quadratic cost function

$$J(\boldsymbol{\alpha}) = \frac{1}{2}\mathbf{b}^T W \mathbf{b} \tag{4.59}$$

$$= \frac{1}{2}\boldsymbol{\alpha}^T R_s^T W R_s \boldsymbol{\alpha} \tag{4.60}$$

subject to a linear inequality constraint and a linear equality constraint, both of which will be discussed below. The term W is a positive semidefinite diagonal weighing matrix, the choice of which will also be discussed in the next section. The minimization of the cost function attempts to keep $B(\tau_i)$ as small as possible for τ_i values different from zero, consistent with the form of the Dirac delta function.

The first constraint is the following inequality:

$$\boldsymbol{\alpha} \geq 0 \tag{4.61}$$

which ensures that the resulting A_α matrix will satisfy the positive simidefinite condition. The second constraint is motivated by the definition of the Dirac delta function, specifically that it integrates to one. If the autocorrelation function of the polynomial process noise is intended to approximate white noise, then the summation in Eq. (4.47) should approximate a Dirac delta function and thus also integrate to one. In the case of a continuous polynomial, that constraint takes the form

$$\int_{-\Delta t}^{\Delta t} \sum_{j=0}^{M} \alpha_{jj} R_{jj}(\tau) d\tau = 1 \tag{4.62}$$

The finite sum approximation of the integral in Eq. (4.62) takes the form

$$\mathbf{c}^T \mathbf{b} = \mathbf{c}^T R_s \boldsymbol{\alpha} = 1 \tag{4.63}$$

where

$$\mathbf{c} = \Delta\tau \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{4.64}$$

In the last equation, $\Delta\tau = \tau_{i+1} - \tau_i = \frac{2\Delta t}{N_s}$ is the sample interval used to construct $\mathbf{b}$ in Eq. (4.55).

Note that in the actual calculations, numerical approximations of $R_{jj}(\tau_i)$ have been used to construct the elements of the R_s matrix in Eq. (4.58). These approximations use numerical integration to evaluate the right-hand side of Eq. (4.49). The number of Euler integration intervals used to compute a typical approximation of $R_{jj}(\tau_i)$ is 401.

4.4.2 Example Development of a Piecewise Polynomial Approximation of White Noise

This subsection designs and analyzes an example piecewise polynomial noise model. It also analyzes the effect of the number of polynomials $M + 1$ on the white-noise approximation accuracy.

The example design is verified by analyzing a simulated realization of the piecewise polynomial noise model. The design involves choosing a set of polynomials $C_j(\eta)$ and the number of polynomials $M + 1$. For the present example, and throughout the remainder of this paper, Chebyshev polynomials have been used. These polynomials, which are easily generated using recursive formulas, have favorable cross-correlation properties and predictable error properties over the interval from -1 to 1.[14] The example number of polynomials has been chosen to be $M + 1 = 11$.

The final part of the design is to determine the vector of scaling factors defined in Eq. (4.57). Recall that this is done by solving the quadratic programming problem presented in the previous subsection. In order to solve that problem, several parameters must be set. The number of samples is chosen to be $N_s = 201$,

and the weighing matrix W is chosen to be a modified version of the $N_s \times N_s$ identity matrix. Like the identity matrix, W has ones along its diagonal, but the center 15 percent of its diagonal elements are set to zero. This heuristically derived weighting penalizes in Eq. (4.59) any deviation from zero towards the ends of the polynomial noise's autocorrelation, but allows deviations towards the middle. Unweighting the middle affords the optimization enough flexibility to satisfy the constraints in Eqs. (4.61) and (4.62) in a manner that helps the autocorrelation approximate the Dirac delta function. The resulting sets of optimal scaling factors for all M values from 0 to 10 are presented in Tables 4.1 and 4.2 for the case $\Delta t = 1$. If Δt is different from 1, then the α values in Tables 4.1 and 4.2 must be rescaled by the factor $\frac{1}{\Delta t_{new}}$. Note that the α values in Tables 4.1 and 4.2 yield slight errors in the normalization constraint in Eq. (4.62), errors of less than 1%. These small errors are attributable to the relatively small value of N_s that has been used to set up the α optimization problems and to the relatively small number of points used to numerically approximate the integrals that determine the R_{jj} autocorrelation functions.

A realization of this process noise can be constructed by generating random polynomial coefficients according the statistics defined in Eqs. (4.43) and (4.44) for a large number of integration intervals. The resulting piecewise continuous noise is analyzed in order to show that its statistics agree with the theory. The following steps have been followed to create a scalar piecewise polynomial noise realization:

1. Choose the continuous-time noise intensity Q in Eq. (4.44). For simplicity, this scalar is set to unity.
2. Generate random polynomial coefficients according to the statistics de-

Table 4.1: Optimized polynomial noise coefficient covariance scaling vectors, α, for $M = 0, 1, \ldots, 5$.

j	M = 0	M = 1	M = 2	M = 3	M = 4	M = 5
0	1.00000000	1.00000000	0.80763605	1.00000000	0.81233857	0.77186939
1		0.35760408	0.00000000	0.00000000	0.19385607	0.00000000
2			1.80266096	0.00000000	1.68713208	2.13783371
3				2.26233225	0.00000000	1.23665530
4					2.03418489	0.00000000
5						2.16598011

fined in Eqs. (4.43) and (4.44) using Q from step 1 and α from Table 4.2. For each interval from t_k to t_{t+1}, $M + 1$ independent scalar coefficients are obtained for this scalar noise example.

3. Densely sample the corresponding realization of the piecewise polynomial noise model. Dense sampling means sampling with a spacing that is much closer than the piecewise polynomial interval Δt.

The resulting sequence is a densely sampled version of $w(t)$, which is plotted over 1,000 piecewise polynomial intervals, each of unit length, in Fig. 4.6. This example uses M = 10, which yields $M + 1 = 11$ piecewise polynomials per interval. Figure 4.7, which shows a close up of the first three intervals, reveals the piecewise polynomial structure. Note that the sample interval $\Delta t = 1$ has been used in this example.

An analysis has been used to verify that the statistics of this realization match the theory that has been used to design it. In particular, the autocorrelation of

Table 4.2: Optimal polynomial noise coefficient covariance scaling vectors, $\boldsymbol{\alpha}$, for $M = 6, 7, \ldots, 10$.

j	M = 6	M = 7	M = 8	M = 9	M = 10
0	0.80835445	0.71434311	0.81179258	0.78044527	0.81610341
1	0.93181255	0.77392921	1.40863993	1.35158950	1.60006409
2	1.72472580	2.63136217	1.69586744	2.00404106	1.65584545
3	0.76752142	0.56510364	1.10715424	1.09078250	1.24788453
4	1.74449507	1.29677485	1.65692286	1.42710078	1.66362196
5	0.00000000	1.50903074	0.60884398	1.32314487	0.84909729
6	1.97400828	0.00000000	1.61190861	0.65538133	1.50507710
7		1.81337200	0.00000000	1.40826626	0.42558707
8			1.54757888	0.00000000	1.28861941
9				1.31627568	0.00000000
10					1.08351061

the densely sampled version of $w(t)$ has been calculated and compared to the theoretical autocorrelation predicted by Eq. (4.47) for the applicable α_{jj} coefficients from Table 4.2. The simulated data's autocorrelation and the theoretical autocorrelation are both shown in Fig. 4.8. Both curves integrate to one over the interval from $-\Delta t$ (= -1) to $+\Delta t$ (= $+1$). The agreement between the curves indicates that the simulated piecewise polynomial noise behaves as expected over a large number of integration intervals.

In addition to analyzing the statistics of the simulated data, it is important to demonstrate the polynomial noise's ability to approximate white noise. This behavior is shown in Fig. 4.9, which plots autocorrelation functions for noise

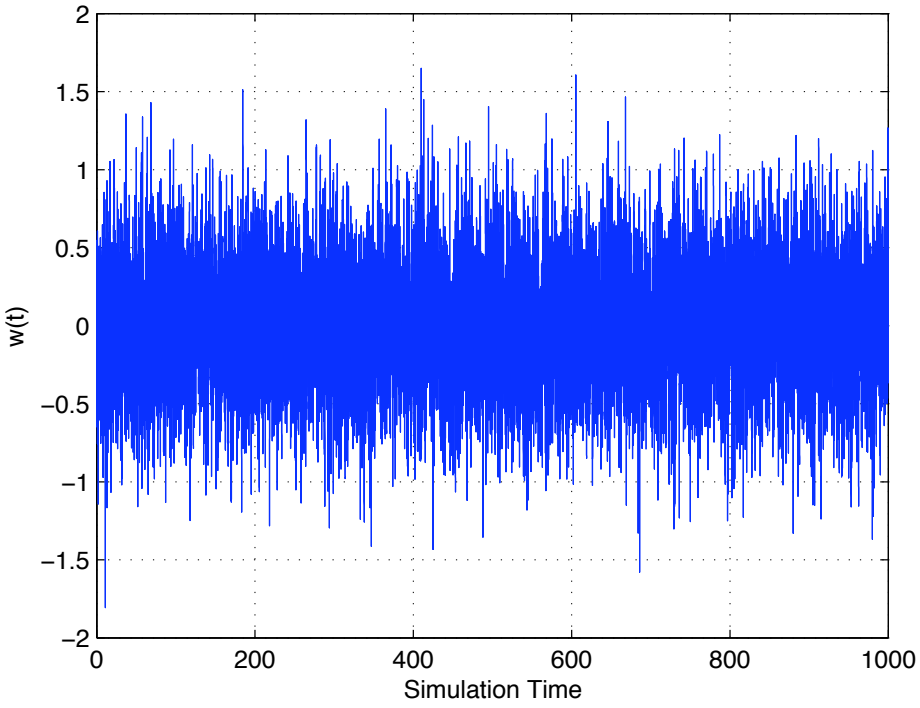

Figure 4.6: The time history of simulated polynomial noise over 1,000 piecewise polynomial intervals.

models with piecewise polynomials of increasing order, ranging from $M = 0$ to $M = 15$. When the piecewise polynomial noise model is realized at $M = 0$, the result is the familiar zero-order hold, or piecewise constant, noise. Its autocorrelation, the low triangular-shaped dashed curve, is the poorest approximation to continuous-time white noise of the four shown in Fig. 4.9, recalling that the target autocorrelation is the Dirac delta function. When the model is realized at $M = 15$, the resulting noise is a better approximation of continuous-time white noise, as is evident in its autocorrelation function. It looks much narrower, like the desired delta function. This behavior demonstrates that as the order of the piecewise polynomial noise increases, it better approximates the behavior of continuous-time white noise. Note that each curve of Fig. 4.9 uses a different

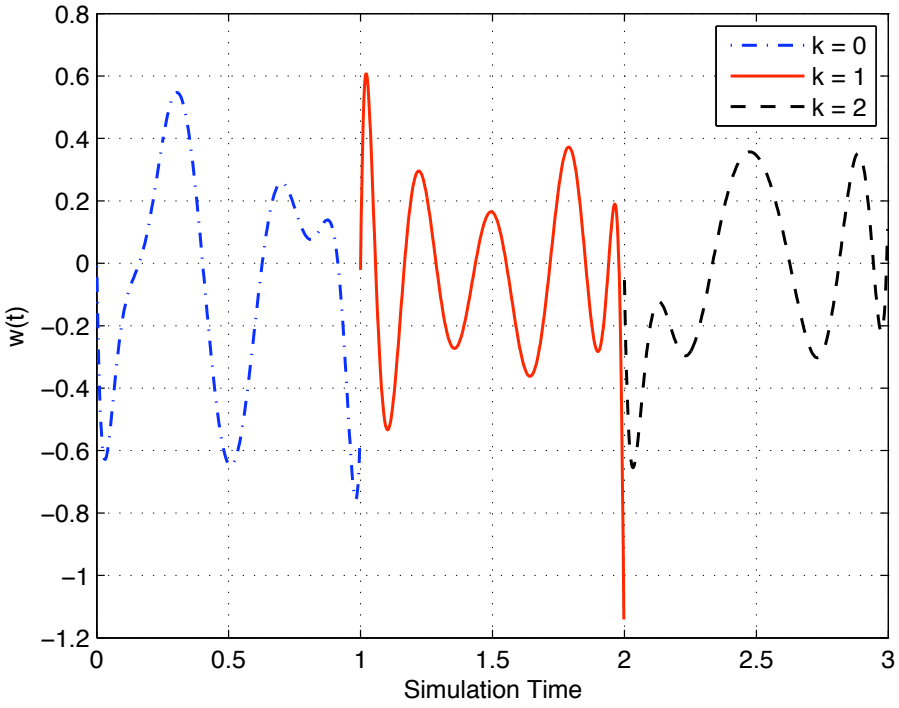

Figure 4.7: The time history of simulated polynomial noise over 3 piecewise polynomial intervals.

set of α_{jj} values as determined by solving a different quadratic program, as per Tables 4.1 or 4.2. In the case of $M = 0$, $\alpha_{00} = \frac{1}{\Delta t}$, as dictated by Eq. (4.62). This demonstrates that the present developments represent a generalization of the standard approximation of continuous-time white noise via a simple zero-order hold model, which is the $M = 0$ model.

Note that the intended use of this process noise model is in the context of a sampled data Kalman filter that uses dense output Runge-Kutta numerical integration, as is discussed in the Appendix. For a given order, the Runge-Kutta method has a limited ability to accurately approximate the effects of rapid, high-order dynamic variations. This fact imposes a limit on the order of the polynomials used in the piecewise polynomial process noise model. A simple

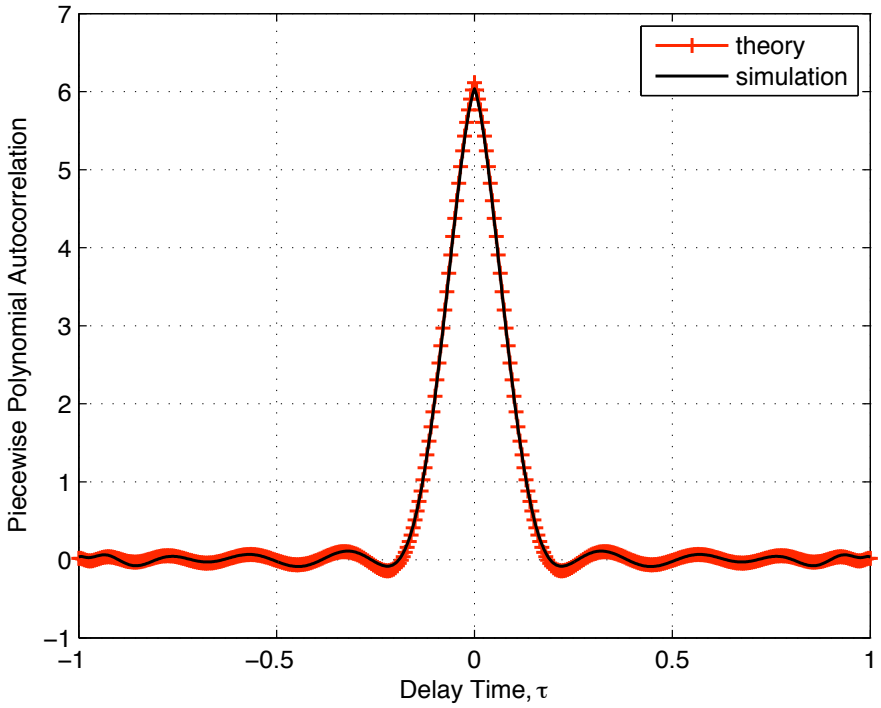

Figure 4.8: Theoretical and simulated piecewise polynomial noise autocorrelation for the case $M = 10$.

analysis suggests that the order of the polynomials, M, should be less than or equal to the order of the Runge-Kutta method minus one. It is conjectured that using too large a value for M for a given Runge-Kutta order does no more harm than the following: useless extra calculations are carried out by the Kalman filter in a vain attempt to simulate white-noise bandwidth that lies outside of the sensitive range of the Runge-Kutta numerical integration.

4.5 Demonstration of the Kalman Filter in a Simulation

The Kalman filter described in Section II that processes implicitly constrained measurements and that incorporates the piecewise polynomial process noise

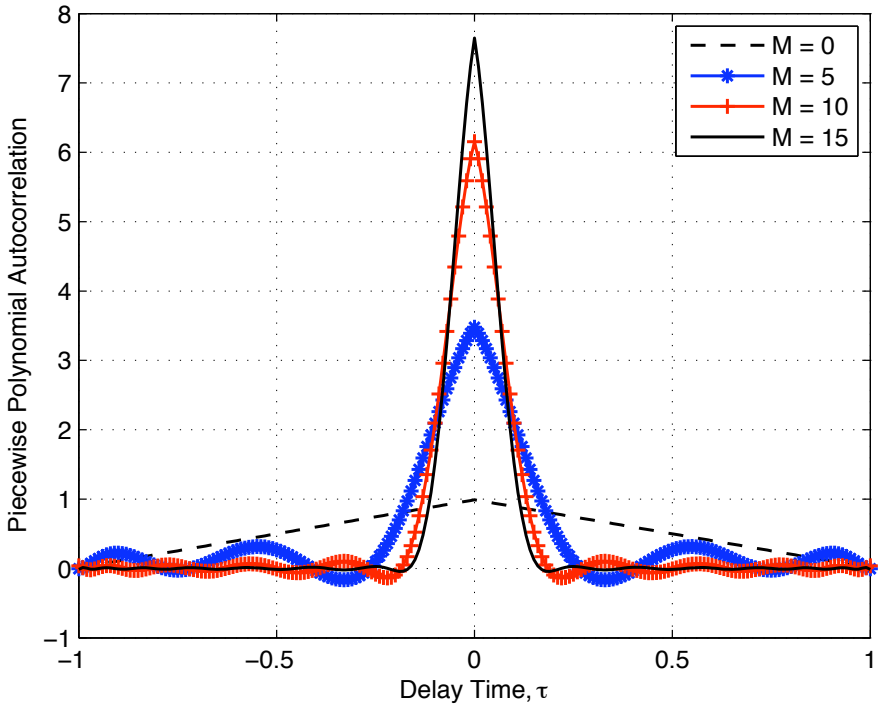

Figure 4.9: Autocorrelation for polynomial noise realizations of different order.

model described in Section III is demonstrated in this section by solving a simple radio navigation tracking problem using data from a truth-model simulation. One goal of the simulation is to test how well the piecewise polynomial process noise theory approximates the true continuous-time process noise statistics. The statistics of the square-root information filter (SRIF) residuals, $\mathbf{z}_{res}$ in Eq. (4.33), are used to investigate this issue. A second goal of the simulation is to test how well the filter performs as the order of the piecewise polynomial process noise model is increased from the standard zero-order hold model and as the number of measurements in a given Kalman filter interval is increased. For this test, the sum of the squares (SOS) of the SRIF residuals will be used as a performance indicator. Finally, the new SRIF will be compared, in terms of estimation accuracy and computational burden, to a traditional SRIF that restarts

its numerical integrator at each of the state prediction iterations.

The example problem involves a vehicle that transmits a radio signal to two stationary receivers, as illustrated in Fig. 4.10. The vehicle is allowed to move

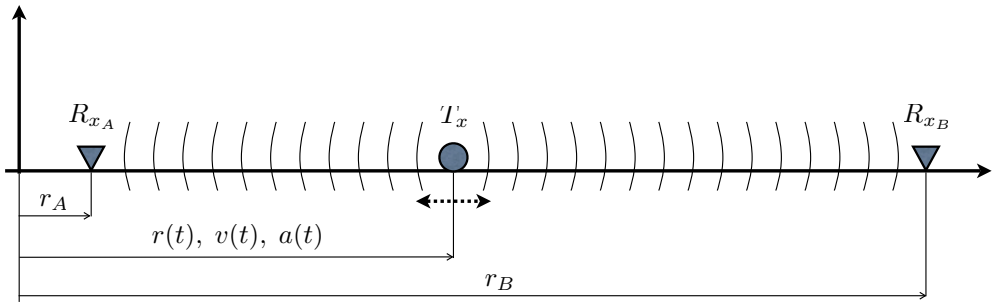

Figure 4.10: Tracking scenario.

along a straight line with three states (position, velocity, and acceleration) according to a continuous-time, triple-integrator dynamics model. The acceleration is driven by a white-noise process. Additionally, the transmitter clock error is modeled as a two-state random process in which both its phase and frequency errors are driven by white-noise processes. The clock model is taken from Ref. 3. The resulting linear, continuous-time system is given in the following vector-matrix equation:

$$\begin{bmatrix} \dot{r}(t) \\ \dot{v}(t) \\ \dot{a}(t) \\ c\dot{\delta t}_{clk}(t) \\ c\dot{\delta f}_{clk}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} r(t) \\ v(t) \\ a(t) \\ c\delta t_{clk}(t) \\ c\delta f_{clk}(t) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} w_a(t) \\ w_t(t) \\ w_f(t) \end{bmatrix} \tag{4.65}$$

where $\delta t_{clk}(t)$ is the transmitter clock phase error, $\delta f_{clk}(t)$ is the transmitter clock

frequency error, c is the speed of light, and $w_a(t)$, $w_t(t)$, and $w_f(t)$ are three uncorrelated zero-mean Gaussian white-noise processes. They have the following autocorrelations:

$$E[w_a(t)w_a(\tau)] = q_{ct}\delta(t-\tau) \tag{4.66}$$

$$E[w_t(t)w_t(\tau)] = c^2\frac{h_0}{2}\delta(t-\tau) \tag{4.67}$$

$$E[w_f(t)w_f(\tau)] = c^2 h_{-2}2\pi^2\delta(t-\tau) \tag{4.68}$$

where q_{ct} is the white-noise intensity for the acceleration perturbation, and where h_0 and h_{-2} are Allan variance parameters for the clock model.

The truth states at the signal transmit times are generated using the equivalent discrete-time model driven by discrete-time white-noise sequences that have the same effect as the continuous-time random processes in Eq. (4.65). Time histories of the true position and transmitter clock phase error are combined with the known receiver locations to generate noisy pseudorange observables according to the following model, which assumes perfect receiver clocks:

$$P_A(t_{A_{k(q)}}) = r(\bar{t}_{k(q)}) - r_A - c\delta t_{clk}(\bar{t}_{k(q)}) + \nu_A(t_{A_{k(q)}}) \tag{4.69}$$

$$P_B(t_{B_{k(p)}}) = r_B - r(\bar{t}_{k(p)}) - c\delta t_{clk}(\bar{t}_{k(p)}) + \nu_B(t_{B_{k(p)}}) \tag{4.70}$$

For receiver A, $P_A(t_{A_{k(q)}})$ is the q^{th} pseudorange measured on the k^{th} filtering interval at the known signal reception time $t_{A_{k(q)}} = \bar{t}_{k(q)} + [r(\bar{t}_{k(q)}) - r_A]/c$, $r(\bar{t}_{k(q)})$ is the transmitter's position at the signal transmit time $\bar{t}_{k(q)}$, which is unknown to the Kalman filter, $c\delta t_{clk}(\bar{t}_{k(q)})$ is the range-equivelant transmitter clock phase error, and $\nu_A(t_{A_{k(q)}})$ is white Gaussian measurement noise. Similar definitions apply to receiver B. An important feature of the scenario design is that the receivers are on opposite sides of the transmitter at all times. This feature guarantees the observability of the position, velocity, and acceleration states along with the clock

phase and frequency error states. The signal transmit times $\bar{t}_{k(q)}$ and $\bar{t}_{k(p)}$, which are unknown to the Kalman filter, are implicitly defined by the following light time equations:

$$0 = g_{A_{k(q)}}[t_{A_{k(q)}}, \bar{t}_{k(q)}, r(\bar{t}_{k(q)})] = r(\bar{t}_{k(q)}) - r_A - c(t_{A_{k(q)}} - \bar{t}_{k(q)}) \tag{4.71}$$

$$0 = g_{B_{k(p)}}[t_{B_{k(p)}}, \bar{t}_{k(p)}, r(\bar{t}_{k(p)})] = r_B - r(\bar{t}_{k(p)}) - c(t_{B_{k(p)}} - \bar{t}_{k(p)}) \tag{4.72}$$

For a particular scenario, the transmitter's position time history is plotted in Fig. 4.11. Note that the transmit times, $\bar{t}_{k(q)}$ and $\bar{t}_{k(p)}$, are explicitly computed in the truth-model simulation and that Eqs. (4.71) and (4.72) are used by the truth-model simulation to compute the signal reception times, $t_{A_{k(q)}}$ and $t_{B_{k(q)}}$. It is only in the filter that Eqs. (4.71) and (4.72) must be used to solve for the transmit times.

Given the dynamics model in Eq. (4.65), the continuous-time statistics in Eqs. (4.66) through (4.68), the sampled measurement model in Eqs. (4.69) and (4.70), and the implicit transmission time constraints in Eqs. (4.71) and (4.72), it is straightforward to develop a truth-model simulation and to design a Kalman filter based on this paper's principles. This has been done. The remainder of this section discusses the results of applying the new filter to the corresponding simulated data.

The SRIF residual statistics offer a good indicator of the fidelity with which the piecewise polynomial process noise model approximates the underlying continuous-time random processes used to generate the truth states. If the filter's model is reasonable, then the SRIF residuals will be samples from a zero-mean white-noise distribution. Equivalently, the SOS of the SRIF residuals at each filtering step should be drawn from a χ^2_n distribution with $n = N_m$ degrees of freedom, where N_m is the number of measurements processed during each

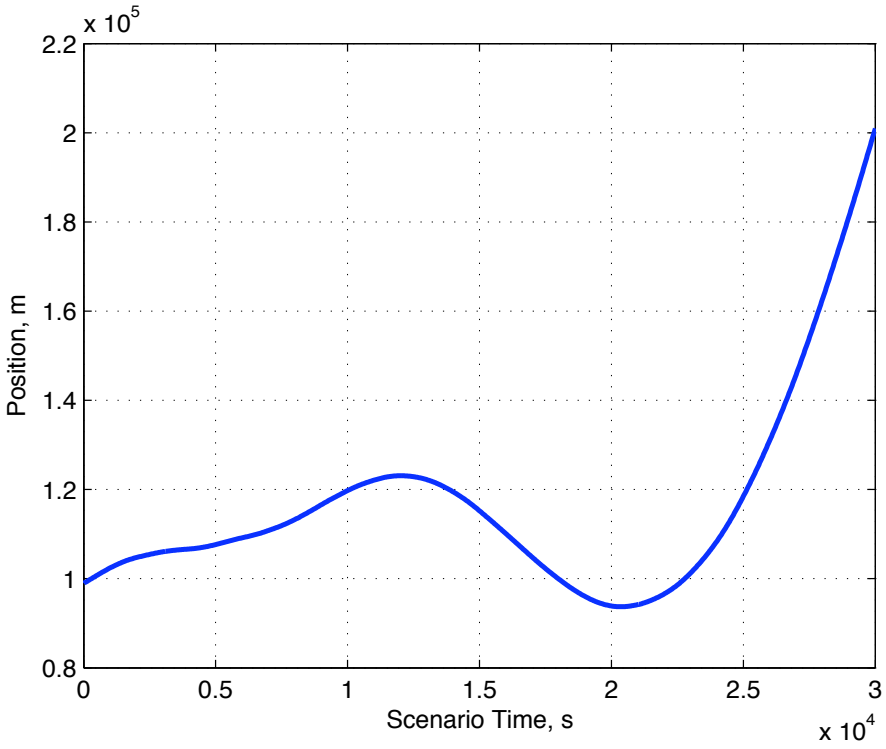

Figure 4.11: Simulated transmitter position time history.

filtering interval. Figure 4.12 shows a time history of the SRIF residuals for an example case. The mean of the residuals over simulation time is -0.0010, and the variance is 1.0402, both indicating a close match to the expected statistics. The autocorrelation function associated with this plot has been computed, and it closely resembles the required Dirac delta function form.

Next, three different simulation scenarios, each with a different number of measurements, N_m, on each filtering interval have been processed by three different filters, each with a different polynomial process noise model order, M. Figure 4.13 shows the normalized SOS of the SRIF residuals for each scenario and each filter. The groups of bars indicate filters of a particular polynomial order, i.e., groups of bars for $M = 0$, 2, and 4. Each bar in a particular group indicates the normalized SOS of the SRIF residuals over a simulation time. The

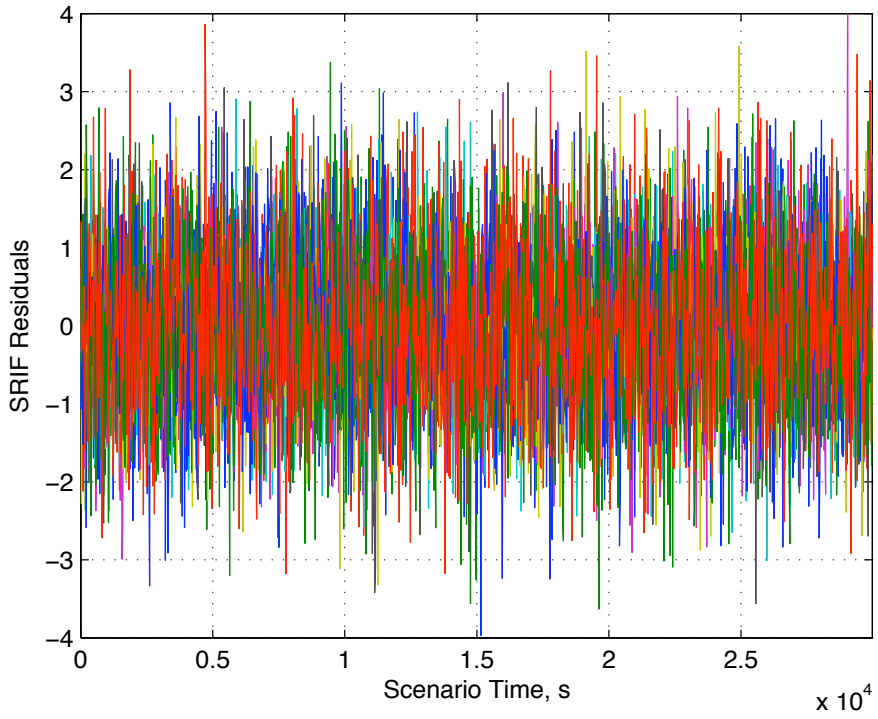

Figure 4.12: The time history of the SRIF residuals for a particular simulation.

SOS values are divided by the total number of measurements processed in the scenario in order to normalize them.

There are several important aspects to this plot. First, it is obvious that in this particular case the magnitude of the normalized SOS both increases and decreases as the number of measurements is increased, i.e., the bars in each group get larger in some cases and smaller in other cases as the number of measurements is increased. Slightly different simulation parameters result in trends that showed all bars increasing or all bars decreasing. This example has been chosen because it reveals some interesting behavior. Since the normalized SOS is a measure of how well the model fits the data, increasing the amount of data will usually result in an increase in SOS since finding a good fit becomes

more challenging. In some circumstances, however, if the model is particularly well suited for a set of measurements, the SOS may indicate an improved fit even as the number of measurements is increased. In the case presented here, both types of behavior are present, revealing that particular polynomial orders may be better suited for particular measurement scenarios. The zero-order and 4^{th}-order models appear to be better suited for the measurement scenarios that contain 2 or 4 measurements than is the 2^{nd}-order model. In all cases, however, the accuracy of the estimate improves due to the extra information that the additional data provides. This claim is supported by the data tabulated in Table 4.3, which shows the computed steady-state estimation error standard deviations for a simulation scenario with 2, 6, and 10 measurements per integration interval. Those standard deviations exhibit notable improvement as the measurement density is increased.

Table 4.3: For $M = 4$, variations of computed estimation error standard deviations with the number of measurements per integration interval, N_m.

N_m	σ_r (m)	σ_v (m/s)	σ_a (m/s^2)	$\sigma_{c\delta t_{clk}}$ (m)	$\sigma_{c\delta f_{clk}}$ (m/s)
2	5.3209e-01	1.4214e-03	1.5859e-06	7.9075e-00	6.9407e-02
6	4.7947e-01	1.2809e-03	1.4291e-06	7.1255e-00	6.2543e-02
10	1.6423e-01	1.7157e-04	7.9398e-08	6.9355e-01	5.1747e-03

A better measure of the filters' performance is to look at their capacity to fit a particular amount of data as the polynomial order is increased. This behavior is revealed by noting, on Fig. 4.13, how the heights of bars associated with a particular number of measurements change as a function of the polynomial order. For example, the relative heights of the left-most bars in each group in-

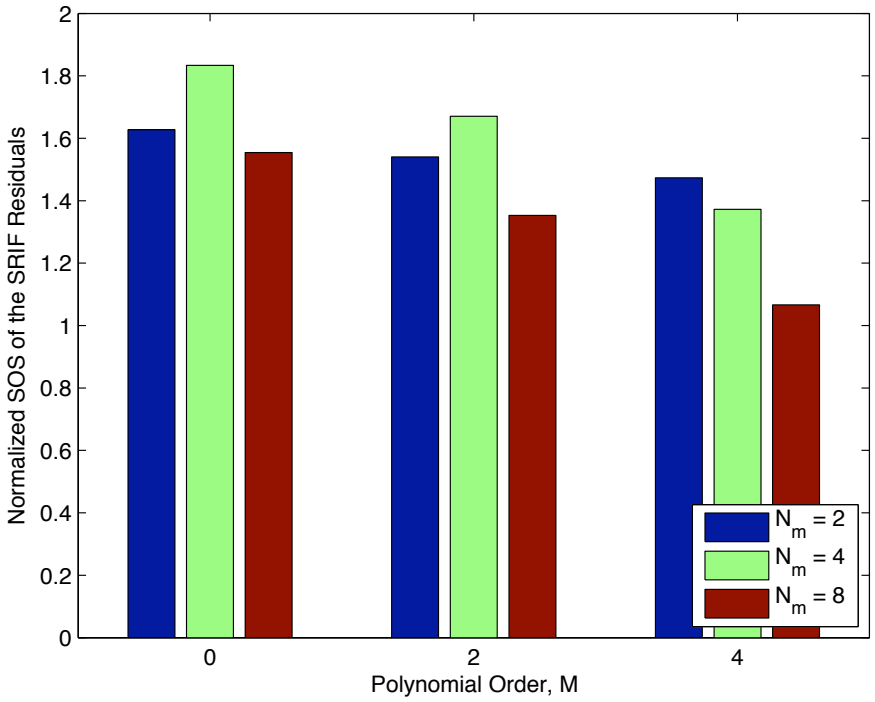

Figure 4.13: The relative effect of the number of measurements per integration interval, N_m, for three different piecewise polynomial process noise models of different order, M.

dicate how the polynomial order changes the filters' ability to fit the data given $N_m = 2$. For all such comparisons, the filters' capacity to fit the data increases as the polynomial order increases. The trend is most pronounced, however, when the measurement density is highest. For a more direct comparison, Fig. 4.14 reorganizes the groupings by the number of measurements rather than the order of the polynomial model, and the color coding of the bars is changed to reflect the order of the polynomial. Thus, each grouping of three bars shows how the fit improves as the polynomial process noise model order M increases for a fixed value of N_m. The downward trend in each grouping as M goes from 0 to 4 is the important feature of this plot. The downward trend is most pronounced for $N_m = 8$ measurements per integration interval, as expected. This trend suggests

that the additional fit parameters offered by the higher-order piecewise polynomial process noise models provide the filter increased capability to fit dense data over a given filtering interval.

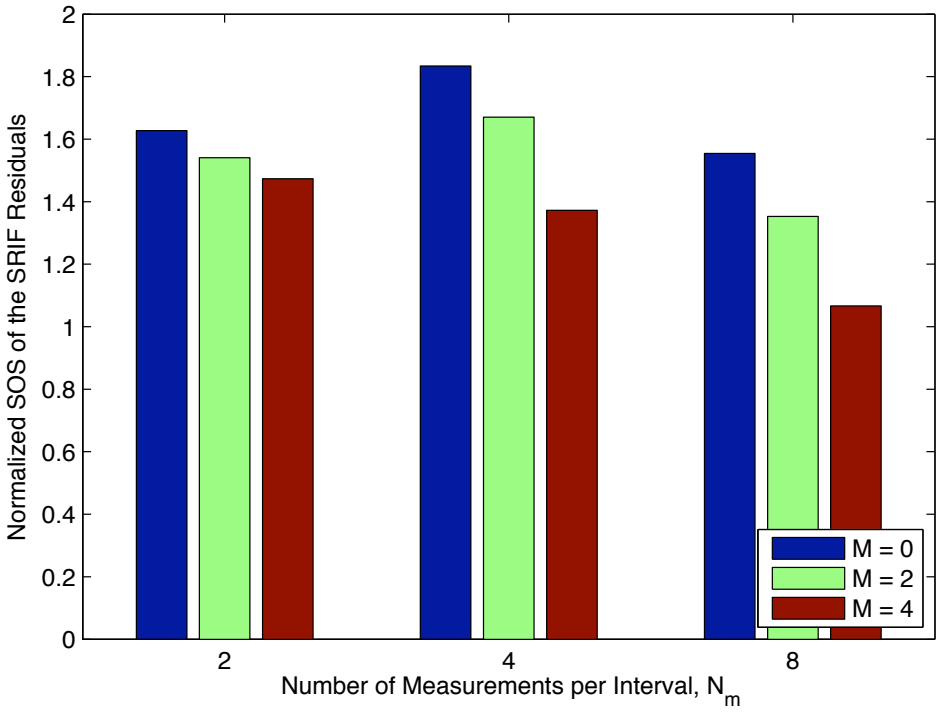

Figure 4.14: The relative effect of the order of the piecewise polynomial process noise, M, for three different numbers of measurements per integration interval, N_m.

In addition to evaluating the new filter's ability to fit the data, its performance is compared to that of a traditional Kalman filter with a zero-order hold process noise model. The traditional filter is effectively the $M = 0$ version of the new filter, with several important differences. First, in order to compute the implicitly defined signal transmit times, $\bar{t}_{k(q)}$, the traditional filter restarts its Runge-Kutta numerical integration for each iteration, re-propagating the state from time t_k to each candidate transmit time as those times and the associated state predictions converge to values that satisfy the transmit time constraint in

Eq. (4.19). Second, the filter processes only one measurement at a time. The converged signal transmit time for the first measurement encountered since the last update, therefore, becomes the terminal time for the Runge-Kutta integration, i.e., $\bar{t}_{k(1)} = t_{k+1}$. This approach effectively defines a variable integration interval, Δt_k. To account for the effects of this variable interval, the zeroth-order noise covariance scaling term must be redefined as $\alpha_{00_k} = 1/\Delta t_k$. Third, by processing one measurement at a time, the filter's linearized measurement model is dependent only on $\Delta\mathbf{x}_{k+1} = \mathbf{x}_{k+1} - \bar{\mathbf{x}}_{k+1}$, not on $\Delta\mathbf{x}_k$ and $\tilde{\mathbf{w}}_k$, thus simplifying the equations used to compute the measurement update. The new filter, however, must still account for the dependence of the estimated signal transmit time on the state estimate computed during the measurement update. It does so by including in the linearized measurement model the effects of the time-of-flight constraint.

The new and traditional filters are compared in terms of their estimation accuracy and computation time. A high measurement density simulation scenario is used to draw these comparisons. The scenario has 1,000 nominal dense output integration intervals that are each 30 seconds long. During each of these intervals, 8 measurements are taken, 4 from receiver A and 4 from receiver B.

The first comparison is in terms of computational burden. As stated earlier, in many filtering problems, the most expensive operation is the evaluation of detailed force models in the numerical integration of the state, the state transition matrix, and the process noise influence matrix. The new method's reliance on a dense output numerical scheme replaces many of the force model function evaluations with much cheaper polynomial evaluations. The relative number of Runge-Kutta integration calls and polynomial evaluations for a zero-order hold process noise filter and a 4^{th}-order polynomial process noise filter are tab-

ulated in the first two rows of Table 4.4. The new filter's 1,000 Runge-Kutta calls, one for each 30-second interval, constitute far fewer than the traditional filter's 17,214 calls. The new filter requires 17,161 polynomial evaluations. The roughly 17,000 Runge-Kutta calls and polynomial evaluations indicate that, in this scenario, the filters usually needed two, and occasionally three, iterations to converge on reasonable estimates of the signal transmit times for each of the 8,000 measurements.

Table 4.4: Number of Runge-Kutta calls and polynomial evaluations and the total computation time spent in propagation.

	Zero-Order Hold	4^{th}-Order
R-K Calls	17,214	1,000
Poly Calls	0	17,161
Time (s)	10.2	2.0

For this test, both methods use the 7-stage, 5^{th}-order Runge-Kutta method described in the Appendix. On the test computer, each Runge-Kutta call took on average 630 microseconds, and each polynomial evaluation took on average 82 microseconds, resulting in 10.2 seconds and 2.0 seconds, respectively, for the total time each algorithm spent computing propagations. It should be noted that since the polynomial approximation for the implemented Runge-Kutta method is 4^{th}-order, and since a shorter effective step size is realized by numerically integrating to each individual measurement, the traditional Kalman filter could achieve comparable integration accuracy with a 4-stage, 4^{th}-order method, saving 3 force model function evaluations on each Runge-Kutta call. The total savings would nearly halve the computation time, still resulting in inferior perfor-

mance to the dense output approach. Another important point to consider is the complexity of the force model. A simple force model was used in this test case. As more complex and computationally burdensome force models are used, the performance difference between the methods will become larger.

Given that the new method requires less computation, the next question to answer is how does its estimation accuracy compare to that of traditional methods. To make this comparison, the steady-state RMS estimation error for a traditional Kalman filter and the new 4^{th}-order polynomial process noise model filter are tabulated in Table 4.5. In this representative case, the new filter slightly outperforms the traditional filter for position, velocity, and acceleration accuracy, and the traditional filter slightly outperforms the new filter for clock bias and drift accuracy. The magnitudes of these differences are all small, suggesting that the performance of the two filters is equivalent for this particular tracking problem.

Table 4.5: Steady-state RMS error for a zero-order hold process noise model filter and a 4^{th}-order polynomial process noise model filter.

Variable	Zero-Order Hold	4^{th}-Order Polynomial
x (m)	0.123	0.0821
v (m/s)	0.00734	0.00513
a (m/s^2)	0.000229	0.000203
$c\delta t$ (m)	0.224	0.278
$c\dot{\delta}t$ (m/s)	0.0558	0.0679

4.6 Conclusions

This paper has presented a new approximation to a continuous-time sampled-data extended Kalman filter that is appropriate for systems with implicitly defined measurement times. The method is useful for any system in which such measurements arise, including many radio navigation and tracking problems. The filter uniquely combines dense output numerical propagation with a finite-dimensional approximation of the underlying continuous-time white process noise. It does so in a joint propagation/measurement update step that can process multiple, implicitly constrained measurements at once. The filter's models have been presented and linearizations of those models have been derived. The combined SRIF dynamic propagation and measurement update that produces a posteriori state and process noise parameter estimates has been explained. A new statistical model for the finite-dimensional approximation to white process noise has been developed and demonstrated in simulation, and the filter has been tested on a simple radio navigation tracking problem by using a truth-model simulation.

The underlying continuous-time white noise has been approximated using a finite-dimensional piecewise polynomial. Each continuous segment of the polynomial represents a randomly weighted sum of polynomial basis functions over a given numerical integration interval. In its simplest form, the polynomial process noise approximation has been shown to be equivalent to a zero-order-hold discrete-time white-noise model. The random polynomial weighting coefficients have been treated as discrete-time random variables that are constant for a particular filtering interval and that are drawn from a zero-mean white-noise sequence. The covariances for these variables have been chosen to be scaled ver-

sions of the continuous-time noise intensity. The scaling factors, which ensure that the piecewise polynomial process noise optimally approximates the behavior of continuous-time white noise, have been obtained by solving a quadratic programming problem. They have been presented for models that use Chebyshev polynomial basis functions with orders $M = 0, 1, \ldots, 10$. Simulated piecewise polynomial process noise has been analyzed, and its statistics have agreed closely with the statistics predicted by the theory. A theoretical analysis has demonstrated that piecewise polynomials of higher order better approximate continuous-time white noise.

A truth-model simulation based on a simple radio navigation tracking problem has tested the performance of the new Kalman filter. An analysis of the SRIF residuals has shown that the new Kalman filter yields a statistically reasonable approximation to the original problem. Further, the simulations have shown that, for a given measurement density, filters with higher-order polynomial process noise models provide better fits to the data. This conclusion has been reached by analyzing the sums of the squares of the SRIF residuals for scenarios with varying measurement density and polynomial order. The new filter's RMS errors have been shown to be comparable to those of a traditional Kalman filter with a zero-order hold process noise model, and it has achieved this performance using less computation time.

Appendix: A Candidate Dense Output Runge-Kutta Method

Runge-Kutta methods are a class of numerical integration routines that only require a dynamics model and an initial state to propagate that state over an

interval. Such methods, which are referred to as single-step methods, are particularly well suited for use in sequential estimation. They propagate a system's state by evaluating its rates according to a dynamics model at several intermediate points over the interval and by combining those function evaluations with the initial state. The result is an approximation of the state at the end of the interval. Consider an example N_{rk}-step explicit Runge-Kutta method defined by the following two equations:

$$\boldsymbol{\xi}_i = \begin{cases} \mathbf{f}[t_k, \mathbf{x}(t_k)] & for \quad i = 1 \\ \mathbf{f}[t_k + c_i \Delta t, \mathbf{x}(t_k) + \Delta t \sum_{j=1}^{i-1} a_{ij} \boldsymbol{\xi}_j] & for \quad i = 2, 3, \ldots, N_{rk} \end{cases} \tag{4.73}$$

$$\mathbf{x}(t_k + \Delta t) \approx \mathbf{x}(t_k) + \Delta t \sum_{i=1}^{N_{rk}} b_i \boldsymbol{\xi}_i \tag{4.74}$$

In Eq. (4.73), $\boldsymbol{\xi}_i$ is the i^{th} function evaluation at the intermediate time $t_k + c_i \Delta t$ and at the intermediate state approximation $\mathbf{x}(t_k) + \Delta t \sum_{j=1}^{i-1} a_{ij} \boldsymbol{\xi}_j$. The specific choices of the intermediate times and state approximations are defined by the coefficients c_i and a_{ij}, respectively. In Eq. (4.74), the final state approximation equals the initial state plus a weighted sum of the function evaluations. The weighting is defined by the parameters b_i and the step size. The particular choices of the parameters in these equations, which define the specific Runge-Kutta method and ensure its level of accuracy, are unimportant to this development and are not discussed here. For a more thorough discussion, please refer to Refs. 10 and 18. The important point here is that the method's only output is the final state approximation.

In some instances, the user not only requires a state approximation at the end of the integration interval but also at intermediate points over the interval. Numerically propagating to intermediate points, however, may be computationally expensive or even numerically ineffectual if the intermediate step

size becomes very small. Dense output Runge-Kutta methods, sometimes referred to as continuous methods, offer a solution to this problem. In addition to outputting the final state approximation, these methods output the function evaluation values, $\boldsymbol{\xi}_i$. When these values are scaled by the step size and by specially designed normalized polynomial coefficients, they represent coefficients for polynomials that describe the state's evolution continuously on the integration interval. The normalized coefficients are designed in conjunction with the method's other parameters such that the resulting polynomials satisfy the method's order condition over the entire integration interval. Importantly, the continuous approximation of the state is obtained cheaply; computing the necessary coefficients requires few additional function evaluations, if any. For example, in addition to the original N_{rk} function evaluations, the dense output version of the method in Eqs. (4.73) and (4.74) may require additional function evaluations of the form

$$\boldsymbol{\xi}_i = \mathbf{f}[t_k + c_i\Delta t, \mathbf{x}(t_k) + \Delta t \sum_{j=1}^{i-1} a_{ij}\boldsymbol{\xi}_j] \qquad for \quad i = N_{rk}+1, N_{rk}+2, \ldots, N_{rk}+\Delta N_{rk} \tag{4.75}$$

where ΔN_{rk} is the (normally small) number of additional function evaluations. The $\boldsymbol{\xi}_i$ values for $i = 1, 2, \ldots, N_{rk} + \Delta N_{rk}$ are used in the following formula to approximate the state at any intermediate point on the integration interval:

$$\mathbf{x}(t_k + \theta\Delta t) \approx \mathbf{x}(t_k) + \Delta t \sum_{i=1}^{N_{rk}+\Delta N_{rk}} b_i^*(\theta)\boldsymbol{\xi}_i \tag{4.76}$$

Here, θ is a continuous variable between 0 and 1, and $b_i^*(\theta)$ is the i^{th} normalized polynomial in θ. These polynomials take the form

$$b_i^*(\theta) = \sum_{j=1}^{N_{do}+1} d_{ij}\theta^{j-1} \qquad for \quad i = 1, 2, \ldots, N_{rk} + \Delta N_{rk} \tag{4.77}$$

where N_{do} is the order of the dense output interpolant, and where the d_{ij} values are the normalized polynomial coefficients discussed earlier.

An important consideration is how these polynomials are used in the extended Kalman filter presented in this paper. The filter not only requires a continuous approximation of the system's state over each integration interval, but also of the state transition matrix, a continuous version of Eq. (4.5), i.e.,

$$\Phi(t, t_k) = \frac{\partial \mathbf{x}(t; \Sigma_k)}{\partial \mathbf{x}_k} \tag{4.78}$$

and the process noise coefficient influence matrix, Eq. (4.15). One way to evaluate these matrices is to analytically differentiate the dense output Runge-Kutta method's polynomials with respect to the initial state, $\mathbf{x}_k$, and the polynomial process noise coefficients, $\tilde{\mathbf{w}}_k$. Another method, the one used in this study, recognizes that the desired matrices are the solutions to the initial value problems defined in Eqs. (4.6) and (4.9) and thus may be approximated by the same dense output Runge-Kutta method used to propagate the system's state. Such an approach makes sense computationally since many or all of the most expensive calculations, i.e., the function evaluations, are already computed for the state propagation. As it turns out, the two methods can be shown to be equivalent if the solutions to the differential equations for $\mathbf{x}(t)$, $\Phi(t, t_k)$, and $\Gamma(t, t_k)$ are computed in one unified set of dense output Runge-Kutta calculations.

The specific dense output Runge-Kutta method used in this study is from Refs. 6 and 7. It is a 6-stage, 5^{th}-order explicit method that requires a 7^{th} stage for the dense output calculations. The coefficients of this method are presented in standard form in Table 4.6. Its normalized polynomial coefficient functions in θ take the following form:

$$\begin{aligned} b_1^*(\theta) = {} & \theta^2(3 - 2\theta)b_1 + \theta(\theta - 1)^2 \\ & - 5\theta^2(\theta - 1)^2(2558722523 - 314030160)/11282082432 \end{aligned} \tag{4.79}$$

Table 4.6: Coefficients for a dense output Runge-Kutta numerical integration method.

i	c_i	a_{ij}					
2	1/5	1/5	0	0	0	0	0
3	3/10	3/40	9/40	0	0	0	0
4	4/5	44/45	-56/15	32/9	0	0	0
5	8/9	19372/6561	-25360/2187	64448/6561	-212/729	0	0
6	1	9017/3168	-355/33	46732/5247	49/176	-5103/18656	0
7	1	35/384	0	500/1113	125/192	-2187/6784	11/84
	b_j	35/384	0	500/1113	125/192	-2187/6784	11/84
	j	1	2	3	4	5	6

$$b_2^*(\theta) = 0 \tag{4.80}$$

$$b_3^*(\theta) = \theta^2(3-2\theta)b_3 + 100\theta^2(\theta-1)^2(882725551 - 15701508\theta)/32700410799 \tag{4.81}$$

$$b_4^*(\theta) = \theta^2(3-2\theta)b_4 - 25\theta^2(\theta-1)^2(443332067 - 31403016\theta)/1880347072 \tag{4.82}$$

$$b_5^*(\theta) = \theta^2(3-2\theta)b_5 + 32805\theta^2(\theta-1)^2(23143187 - 3489224\theta)/199316789632 \tag{4.83}$$

$$b_6^*(\theta) = \theta^2(3-2\theta)b_6 - 55\theta^2(\theta-1)^2(29972135 - 7076736\theta)/822651844 \tag{4.84}$$

$$b_7^*(\theta) = \theta^2(\theta-1) + 10\theta^2(\theta-1)^2(7414447 - 829305\theta)/29380423 \tag{4.85}$$

Notice how the coefficients of these functions ensure that when $\theta = 0$, $b_i^* = 0$ for all i, and that when $\theta = 1$, $b_i^* = b_i$ for all i, as is expected for consistency between the standard and dense output. The polynomial interpolation function that results from using these coefficient functions in Eq. (4.76) is a 4^{th}-order interpolant; that is, its errors are of the order $(\Delta t)^5$ for $0 < \theta < 1$.

CHAPTER 5

CONCLUSIONS

New estimation algorithms with applications in orbit determination and tracking problems have been developed and presented. The first two papers have addressed the problem of high-altitude relative navigation using carrier-phase differential GPS techniques. They have contributed a new method of integrating integer ambiguity resolution factorization techniques into a square-root information estimator, a robust carrier-phase cycle slip detection and recovery algorithm, new data processing schemes for CDGPS measurements that do not rely on a priori data processing, and detailed dynamics and measurement models for a high-altitude CDGPS extended Kalman filter. The third paper has addressed the problem of implicitly constrained radio navigation measurements. It has contributed a new piecewise polynomial process noise model that is combined in an extended Kalman filter with dense output Runge-Kutta numerical integration in a joint dynamic propagation/measurement update data processing scheme.

BIBLIOGRAPHY

[1] Y. Bar-Shalom, X.R. Li, T. Kirubarajan, and J. Wiley. *Estimation with Applications to Tracking and Navigation*, pages 179–196. Wiley New York, 2001.

[2] G.J. Bierman. *Factorization Methods for Discrete Sequential Estimation*, pages 113–133, 211–214. Academic Press, 1977.

[3] R.G. Brown and P.A.C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering with Matlab Exercises and Solutions*, pages 428–432. John Wiley & Sons, Inc., 3rd edition, 1997.

[4] F. D. Busse, J. P. How, and J. Simpson. Demonstration of Adaptive Extended Kalman Filter for Low-Earth-Orbit Formation Estimation Using CDGPS. *Navigation: Journal of the Institute of Navigation*, 50(2):79–94, 2003.

[5] S. Curtis. Magnetospheric Multiscale Mission...Resolving Fundamental Processes in Space Plasmas. *NASA, MMS Documents*, NASA/TM-2000-209883, 1999.

[6] J.R. Dormand and P.J. Prince. A family of embedded runge-kutta formulae. *Journal of Computational and Applied Mathematics*, 6(1):19–26, 1980.

[7] J.R. Dormand and P.J. Prince. Runge-kutta triples. *Journal of Computational and Applied Mathematics*, 12A(9):1007–1017, 1986.

[8] P.E. Gill, W. Murray, and M.H. Wright. *Practical Optimization*, pages 37–41. Academic Press, 1981.

[9] W. Gurtner. RINEX-2.10 Format. *http://www.ngs.noaa.gov/CORS/instructions2*, Retrieved on 29 January 2005.

[10] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations*, pages 143–154, 188–194, 356–357. Springer, 1993.

[11] R. Kroes and O. Montenbruck. High Accuracy Kinematic Spacecraft Relative Positioning Using Dual-Frequency GPS Carrier Phase Data. *Proceedings of the 2004 ION NTM Conference*, pages 607–613, San Diego, CA, 26-28 January 2004.

[12] R. Kroes, O. Montenbruck, W. Bertiger, and P. Visser. Precise GRACE Baseline Determination Using GPS. *GPS Solutions*, 9(1):21–31, April 2005.

[13] S. Leung and O. Montenbruck. Real-Time Navigation of Formation-Flying Spacecraft Using-Global-Positioning-System Measurements. *Journal of Guidance, Control, and Dynamics*, 28(2):226–235, 2005.

[14] J.C. Mason and D.C. Handscomb. *Chebyshev Polynomials*, pages 1–36. Chapman and Hall CRC, New York, 2003.

[15] S. Mohiuddin and M.L. Psiaki. High-Altitude Spacecraft Relative Navigation Using Carrier-Phase Differential Global Positioning System Techniques. *Journal of Guidance, Control, and Dynamics*, 30(5):1427–1436, 2007.

[16] O. Montenbruck, T. Ebinuma, E.G. Lightsey, and S. Leung. A Real-Time Kinematic GPS Sensor for Spacecraft Relative Navigation. *Aerospace Science and Technology*, 6:435–449, 2002.

[17] O. Montenbruck and E. Gill. State Interpolation for On-Board Navigation Systems. *Aerospace Science and Technology*, 5(3):209–220, 2001.

[18] O. Montenbruck and E. Gill. *Satellite Orbits: Models, Methods, and Applications*, pages 117–134. Springer, New York, corrected 3rd edition edition, 2005.

[19] M.L. Psiaki. FFT-Based Acquisition of GPS L2 Civilian CM and CL Signals. *Proceedings of the 2004 ION GNSS Conference*, pages 457–473, Long Beach, CA, 21-24 September 2004.

[20] M.L. Psiaki. Block Acquisition of Weak GPS Signals in a Software Receiver. *Proceedings of the 2001 ION GPS Conference*, pages 2838–2850, Salt Lake City, UT, 11-14 September 2001.

[21] M.L. Psiaki and H. Jung. Extended Kalman Filter Methods for Tracking Weak GPS Signals. *Proceedings of the 2002 ION GPS Conference*, pages 2539–2553, Portland, OR, 24-27 September 2002.

[22] M.L. Psiaki and S. Mohiuddin. Global Positioning System Integer Ambiguity Resolution Using Factorized Least-Squares Techniques. *Journal of Guidance, Control, and Dynamics*, 30(2):346–356, 2007.

[23] M.L. Psiaki and S. Mohiuddin. Modeling, Analysis, and Simulation of GPS Carrier-Phase for Spacecraft Relative Navigation. *Journal of Guidance, Control, and Dynamics*, 30(6):1628–1639, 2007.

[24] M.L. Psiaki and S. Mohiuddin. Relative Navigation of High-Altitude Spacecraft Using Dual-Frequency Civilian CDGPS. *Proceedings of the 2005 ION GPS Conference*, pages 1191–1207, Long Beach, CA, 13-16 September 2005.

[25] Spirent Federal Systems. GPS Simulator Products Overview. *http://www.spirentfederal.com*, Retrieved on 29 January 2005.

[26] B.D. Tapley, B.E. Schutz, and G.H. Born. *Statistical Orbit Determination*, pages 159–230. Elsevier Academic Press, New York, 2004.

[27] P.J.G. Teunissen, P. Joosten, and D. Odijk. The Reliability of GPS Ambiguity Resolution. *GPS Solutions*, 2(3):62–69, 1999.

[28] J. Wolfe and J.L. Speyer. Effective Estimation of Relative Positions in Orbit Using Differential Carrier-Phase GPS. *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Paper AIAA-2004-4777, Providence, RI, 16-19 August 2004.